Lorenzo Grassi

Cryptanalysis of AES-like Ciphers and Reviving Old Design Ideas for New Constructions

PhD Thesis
Supervised by Christian Rechberger

SCIENCE ▪ PASSION ▪ TECHNOLOGY

TU Graz

Lorenzo Grassi

Cryptanalysis of AES-like Ciphers and Reviving Old Design Ideas for New Constructions

Doctoral Thesis

to achieve the university degree of
Doktorin der technischen Wissenschaften

submitted to
Graz University of Technology

Advisor: Christian Rechberger

Assessors: Christian Rechberger
Graz, University of Technology (Austria)

Anne Canteaut
Inria, Paris (France)

Institute of Applied Information Processing and Communications
Graz University of Technology

Graz, April 2019

Abstract

In this thesis, I present the research I did with my co-authors on several aspects of symmetric cryptography from September 2015 to April 2019, that is, when I was a PhD student at IAIK, Graz University of Technology (Austria) under the supervision of Christian Rechberger. My research has spanned two different areas of symmetric cryptography, that is the cryptanalysis of existing symmetric ciphers and the design of new ones.

After a brief introduction to block ciphers and their cryptanalysis, the first part of this thesis concerns my work on *cryptanalysis of (round-reduced) AES* (and AES-like ciphers). Usually, the security of symmetric cryptographic primitives cannot be proven. Hence, an important part of symmetric cryptography is cryptanalysis. AES is probably the most used and studied block cipher, and it is not a surprise that there is a vast amount of cryptanalysis on AES. This part contains new approaches to cryptanalyze round-reduced AES. As our main results, we present new properties for up to 5-round AES which are independent of the secret key, improving over a 20 year old result on 4 rounds. Such properties can be used to set up secret- and open-key distinguishers for AES, or they can serve as starting point for new key-recovery attacks. These techniques include: the multiple-of-n distinguisher, mixture differential cryptanalysis and new truncated differential distinguishers based on the mean and on the variance. Besides that, we present new key-recovery attacks on AES with a single secret S-Box, and several observations and new results about open-key distinguishers on AES in the single key-setting.

Modern cryptography developed many techniques that go well beyond solving traditional confidentiality and authenticity problems in two-party communication. The second part of this thesis is devoted to our work on *new designs (strategy) of block ciphers and permutations* which are targeted for new applications in e.g. Multi-Party computation (MPC) and Zero-Knowledge applications. This is motivated by recent progress in these new applications, where primitives from symmetric cryptography are needed and where the cost metric is different from the tradition one in which linear and non-linear computations have (almost) the same cost. Our first contribution here is the development and analysis of a new cipher (called MiMC) with low multiplicative complexity, which resembles a cipher proposed by Knudsen and Nyberg in 1995. As a generalization of such design and of Partial-SPN ciphers in general, we propose a high-level design approach for cryptographic (keyed/unkeyed) permutations – called Hades – addressing both needs of new applications that emphasize the role of non-linear operations and at the same time with a focus on simple arguments for its security. The design is mainly built up on the Wide-Trail design strategy for SP-Networks. At the same time, the crucial feature of such design – that was so far not exploited in details – is of moving from an even to a highly uneven distribution of non-linearity. For our concrete instantiations of HadesMiMC, we borrow ideas from the pre-predecessor of AES, namely SHARK, an S-Box-based design with a single large MDS layer covering the whole internal state, proposed by Rijmen, Daemen, Preneel, Bosselaers and De Win in 1996.

Contents

1 Introduction

Cryptography or cryptology (from Ancient Greek: *kryptós* “hidden, secret” and *graphein* “to write”, or *-logia* “study” respectively) is the branch of science concerned on developing methods that enable secure communication over insecure channels.

Classically, a communication can be considered secure if it satisfies (at least) one of the following three characteristics: confidentiality, integrity and authenticity. Confidentiality refers to the impossibility of a stranger to obtain any meaningful information from the communication that she intercepts. Integrity describes the assurance that the information has not been modified during transmission. Authenticity is the ability to prove the source of a certain information.

In the first part of this thesis, we mainly focus on the problem regarding confidentiality. To provide it, the idea is to convert ordinary information (called “plaintext”) into unintelligible text (called “ciphertext”). The method or algorithm that does such procedure is called a cipher. Using a cipher, a sender can encrypt information and send the encrypted information to a recipient. The recipient can then again use the cipher to decrypt and receive the original information. The detailed operation of a cipher is controlled both by the algorithm and in each instance by a secret “key”.

To set up a confidential communication, it is possible to use symmetric cryptography and/or asymmetric (or public) one. In the first (and older) case, the same key is used for both encryption and decryption. This is in contrast to public key cryptography, where different but related keys are used for encryption and decryption.

Here we will concentrate on symmetric ciphers, in particular on block ciphers. Roughly speaking, a block cipher is a family of permutations indexed by a key. The size of such a key determines the resilience of the cipher to the most basic attack: brute-force. It consists simply in checking all possible keys until the correct one is found. Even if this attack works for every possible cipher (independently of its details), other attacks can be potentially set up as well. Exploiting the mathematical details of the cipher, it can indeed be possible to set up attacks which are more competitive in term of computational cost, but which usually require more data. The branch of cryptography that evaluates the security of (symmetric) schemes and primitives is called cryptanalysis. Since the security of a symmetric scheme can not be mathematical proven, cryptanalysis is never finished, and every year new attack, techniques and scenarios are proposed.

Besides confidentiality, integrity and authenticity, new goals of cryptography have recently emerged, in which block ciphers can play a central role. In the second part of this thesis, we mainly focus on the design of cryptographic (keyed/unkeyed) permutations for several applications, like Multi-Party Computation (MPC) – where the goal is of creating methods for parties to jointly compute a function over their inputs while keeping those inputs private, Zero-Knowledge proof – where a prover can prove that she knows a certain information to a verifier, without communicating any other information other than the fact that she knows it, and many others.

Contributions and Overview of the Thesis

This thesis contains a collection of publications which treat various aspects of block cipher security.

In Part I, we analyze the security of AES (Advanced Encryption Standard) [DR02b], probably the most used and studied block cipher. As main results, we propose new cryptanalysis techniques (which are general enough to be applied to any AES-like cipher) and we show new proprieties and attacks on round-reduced AES.

In Part II, we develop new design strategies for block ciphers. This is motivated by recent progress in practical applications of secure Multi-Party Computation (MPC), Zero-Knowledge (ZK) proofs and Post-Quantum (PQ) Signature schemes, where primitives from symmetric cryptography are needed and where the cost metric is different from the tradition one in which linear and non-linear computations have (almost) the same cost (roughly speaking, linear computations are – compared to non-linear operations – essentially free).

Part I: Cryptanalysis of AES

Cryptanalysis. Block ciphers are certainly among the most important cryptographic primitives. Their design and analysis are well advanced, and with today's knowledge designing a secure block cipher is a problem that is largely considered solved.

Since it is not possible to prove mathematically the security of a symmetric scheme, the security of symmetric cipher is always security against specific attacks. The number of available attacks has increased significantly ever since the introduction of linear [Mat93; Mat94] and differential cryptanalysis [BS90; BS91; BS93] in the early 1990. Besides the numerous variations of linear and differential attacks (e.g. truncated differentials [Knu94], impossible differentials [BBS99; Knu98], and differential-linear cryptanalysis [LH94; BLN14; BLN17] to name only a few), other attacks are based on algebraic properties (as the interpolation attack [JK97]) or exploit some structural properties (as the integral attack [DKR97; KW02], and its recent generalization, the division property [Tod15b]). The consequence of this is that, even if a cipher is today considered secure, it can be potentially broken in the future by e.g. new cryptanalysis techniques. A recent and concrete example of this is the case of MISTY1 [Mat97], a block cipher designed in 1995 by Mitsuru Matsui and others for Mitsubishi Electric and standardized by projects, such as CRYPTREC, ISO/IEC, and NESSIE. Even if it was designed based on the theory of provable security against differential and linear attacks, at CRYPTO 2015 Todo presented the first key-recovery attack on full MISTY1 [Tod15a; Tod17], based on the division property technique proposed by him some time before at EUROCRYPT 2015.

Another important aspect to keep in mind is that the attacker model is regularly changing. With the introduction of statistical attacks, especially linear and differential cryptanalysis, the attacker was suddenly assumed to be able to retrieve, or even choose, large amounts of plaintext/ciphertext pairs. Later, in the related-key setting, the attacker became even more powerful and was assumed to be able to choose not only plaintexts but also ask for the encryption of chosen messages under a key that is related to the unknown secret key. Finally, in the open-key model, the attacker either knows the key or has the ability to choose the key herself.

While the practical impact of such models is often debatable, they actually might become meaningful when the block cipher is used as a building block for other primitives, in particular for the construction of hash-functions. Moreover, even if those considerations do not pose practical attacks, they still provide very useful insights and observations that strengthen our understanding of block ciphers

Cryptanalysis of AES. The Advanced Encryption Standard (AES) [DR02b] is the best known and most widely used secret key cryptosystem, and determining its security is one of the most important problems in cryptanalysis. Since no known attack can break the full AES significantly faster than via exhaustive search, researchers had concentrated on attacks which can break reduced round versions of AES. While such attacks do not pose any practical threat to the AES, they give new insights in the cipher that is probably responsible for the largest fraction of encrypted data worldwide.

Such attacks are important for several reasons. First of all, they enable us to assess the remaining security margin of AES, defined by the ratio between the number of rounds which can be successfully attacked and the number of rounds in the full AES. In addition, there are many proposals for using reduced round AES (and especially its 4 or 5 rounds versions) as components in larger schemes, and thus successful cryptanalysis of these variants can be used to attack those schemes. Only to give some examples, such proposals include Simpira [GM16], ZORRO [GGNS13], LED [GPPR11], Haraka [KLMR16], and AES-PRF [MN17] among many others.

Finally, new cryptanalysis techniques can enable us to develop new attack strategies which may become increasingly potent with additional improvements. In most of the cases, it took several years - and a series of subsequent improvements - from the invention of the technique until it was developed into its current form. As a concrete example, consider the impossible differential cryptanalysis on AES. When it was proposed in 2001 by Biham and Keller [BK01], the impossible differential attack could attack ("only") 5 rounds of AES and it was not competitive with respect to others attacks, as the integral one. It took approximately 6 years before that attack was extended and set up against 7-round AES-128 [Pha04], becoming one of the few attacks on such number of rounds. Finally, only recently Boura *et al.* [BLNS18] improved it into its best currently known variant which breaks 7-round AES with an overall complexity of about 2^{107}.

Cryptanalysis of AES - Our Contribution. Rouglhy speaking, in the last recent years cryptanalysis has mainly focused on maximizing the number of rounds that can be broken without exhausting the full codebook and key space. The most successful attacks often become de-facto standard methods of cryptanalysis for a particular block cipher. In many cases, this process often leads to attacks marginally close to that of pure brute force.

Here we consider a different point of view. Instead of focusing/improving already existing attacks, we try to propose *new* methods of cryptanalysis. Even if such new methods can only break or distinguish a number of rounds that is smaller than the ones already present in the literature and/or even if such methods can be less competitive than existing attacks, such new directions in cryptanalysis can be interesting from a research point of view in order to better understand the ciphers that are in used. Moreover, we can not exclude a priori that, when such methods reached their full potential, they can beat the existing attacks.

First of all, in Chapter 4, we introduce the "subspace trail notation", as a generalization of the invariant subspace attack [LAAZ11; LMR15]. While invariant subspace cryptanalysis relies on iterative subspace structures, our analysis is concerned with trails of different subspaces. With this more generic treatment of subspaces, the resulting method is as such a potentially more powerful attack vector. Interestingly, a strong relation exists between subspace trails and (impossible) truncated differential cryptanalysis. In other words, subspace trail turns out to be an *alternative notation* that can be exploited to formally describe several attacks in the literature. While an alternative representation of a cipher can obviously be regarded in itself neither as a design nor as a cryptanalysis result, we remark that the simplicity of a new representation of a cipher can play a significant heuristic role in the investigation of distinguishers and key-recovery attacks. Here we also present "weak-key subspace trails" for AES, that is subspace trails that hold for a class of weak-keys only. They will be the starting point in order to set up chosen-key distinguishers on AES.

In Chapter 5, we propose a precise theoretical analysis of truncated differentials for 5-round AES. Since the development of cryptanalysis of AES and AES-like constructions in the late 1990s, the set of inputs (or a subset of it) which differ only in one diagonal has special importance. It appears in various (truncated) differential, integral, and impossible differential attacks, among others. Here, given a diagonal set 2^{32} plaintexts which differ only in one diagonal, we study the probabilistic distribution of the number of different pairs of ciphertexts that lie in certain subspaces after 5 rounds of AES - denoted as "number of collisions" in the following. For the first time, we are able to show that independently of the secret key;

- the number of collisions is always a multiple of 8 with prob. 1;

- the number of collisions is on average (a little) bigger compared to the case in which the ciphertexts are generated by a random permutation;
- besides the mean, also the variance of such a distribution is (much) higher than for a random permutation.

To show and prove these, we have developed new theoretical approaches. Practical implementations and verification confirm our analysis.

Such results can be exploited to set up new secret-key distinguishers - e.g. the *"multiple-of-n" distinguisher* and the *first truncated differential distinguisher based on the variance*, besides new key-recovery attacks. We remark that *these are the first secret-key distinguishers on 5-round AES which are independent of the secret key, improving over a 20 year old result on 4 rounds.* However, we start to point out that several problems that concern these new results are still open for future research.

At first it was not clear whether the "multiple-of-8" property/distinguisher could at all lead to attacks on AES which are competitive with respect to previously known results. In Chapter 6, we partially resolve this question, by developing a new type of distinguishers and attacks - called "*mixture differential cryptanalysis*" - on round-reduced AES-like ciphers, a way to translate the (complex) "multiple-of-8" 5-round distinguisher into a simpler and more convenient one (though, on a smaller number of rounds). Given a pair of chosen plaintexts, the idea is to construct new pairs of plaintexts by mixing the generating variables of the original pair of plaintexts. Here we theoretically prove that for 4-round AES the corresponding ciphertexts of the original pair of plaintexts lie in a particular subspace if and only if the corresponding pairs of ciphertexts of the new pairs of plaintexts have the same property. As a slight result, we exploit this fact to set up the first (but non-competitive) key-recovery attack on 6-round AES based on a secret-key distinguisher on 5-round AES which is independent of the secret key.

In Chapter 7, we analyze the security of the cipher that is derived from the AES by replacing the S-Box with a secret 8-bit S-Box, while keeping everything else unchanged. This problem has been already considered in the literature - see [BS01; BS10] and [TKKL15]. In those papers, the proposed attacks consist of two steps: first the attacker determines the secret S-Box up to additive constants, then she uses this knowledge and applies key-recovery attacks present in the literature to derive the whitening key. Here we show that another strategy is also possible. Exploiting particular properties of the MixColumns matrix, we show that the attacker is able to deduce information about the whitening key without discovering and/or exploiting any information of the (secret) S-Box.

Finally, in Chapter 8 we study the security of AES in the open-key model, that is in a scenario in which the adversary is assumed to have a full control over the key. Such attacks makes sense in a hash setting, since in practice the attacker has full access and control over the internal computations[1]. The most recent approach to construct a known-key distinguisher for AES has been proposed by Gilbert at ASIACRYPT 2014 [Gil14]. Such a distinguisher considers 8 core rounds, and extends it by one round in each direction, covering full AES-128. As a first contribution, we disprove both conjectures made there to support such result, e.g. showing that - under the assumptions made in [Gil14] - known-key distinguishers on 12 rounds of AES are also possible. As second contribution, we present the *first chosen-key distinguisher on full AES-128 and on full AES-256 in the single-key setting*, by exploiting and combining our "multiple-of-8" distinguisher and the AES invariant subspace trail. These largely improve all the AES chosen-key distinguishers present in the literature, usually proposed in the related-key setting.

Publication List. This first part includes contributions published as part of the following papers at FSE/ToSC 2017, EUROCRYPT 2017, CT-RSA 2018 and Tosc/FSE 2019, as :

[1] We recall that block ciphers and hash functions are very close cryptographic primitives, as the latter can be built from the former and vice-versa.

- L. Grassi, C. Rechberger, and S. Rønjom. Subspace Trail Cryptanalysis and its Applications to AES. In: IACR Trans. Symmetric Cryptol. 2016.2 (2016), pp. 192–225. DOI: 10.13154/tosc.v2016.i2.192-225.

- L. Grassi, C. Rechberger, and S. Rønjom. A New Structural-Differential Property of 5-Round AES. In: Advances in Cryptology – EUROCRYPT 2017. Ed. by J. Coron and J. B. Nielsen. Vol. 10211. LNCS. Springer, 2017, pp. 289–317. DOI: 10.1007/978-3-319-56614-6_10.

- L. Grassi and C. Rechberger. New and Old Limits for AES Known-Key Distinguishers. Cryptology ePrint Archive, Report 2017/255. In Submission. 2017. URL: https://eprint.iacr.org/2017/255.

- L. Grassi. MixColumns Properties and Attacks on (Round-Reduced) AES with a Single Secret S-Box. In: Topics in Cryptology - CT-RSA 2018. Ed. by N. P. Smart. Vol. 10808. LNCS. Springer, 2018, pp. 243–263. DOI: 10.1007/978-3-319-76953-0_13.

- L. Grassi. Mixture Differential Cryptanalysis: a New Approach to Distinguishers and Attacks on round-reduced AES. In: IACR Transaction on Symmetric Cryptology 2018.2 (2018), pp. 133–160. DOI: 10.13154/tosc.v2018.i2.133-160. URL: https://doi.org/10.13154/tosc.v2018.i2.133-160.

- L. Grassi. Mixture Differential Cryptanalysis and Structural Truncated Differential Attacks on round-reduced AES. Cryptology ePrint Archive, Report 2017/832. https://eprint.iacr.org/2017/832. 2017.

- L. Grassi and C. Rechberger. New Rigorous Analysis of Truncated Differentials for 5-round AES. Cryptology ePrint Archive, Report 2018/182. https://eprint.iacr.org/2018/182. 2018.

- L. Grassi, G. Leander, C. Rechberger, C. Tezcan, and F. Wiemer. Weak-Key Subspace Trails and Applications to AES. In Submission. 2018.

Part II: Novel Designs: MiMC and its Generalizations

Modern cryptography developed many techniques that go well beyond solving traditional confidentiality and authenticity problems in two-party communication. Among them, we mention Secure Multi-Party Computation (MPC), Zero-Knowledge proofs (ZK), Fully Homomorphic Encryption (FHE), and many others. In various applications of these schemes, part of the circuit or function that is being evaluated is actually a cryptographic primitive such as a Pseudo-Random Function (PRF), a symmetric encryption scheme or a collision resistant function. In particular, in many cases the performances of such a schemes depend on the underlying symmetric cryptographic primitive. In this second part of the Thesis, we propose new designs which aim to improve the performances of a large class of applications where the total number of field multiplications in the underlying cryptographic primitive poses the largest performance bottleneck.

In Chapter 10, we present "MiMC"[2], a block cipher that resembles $\mathcal{PURE}$. Such design was initially introduced by Knudsen and Nyberg [NK95] from 1995, which was aimed to demonstrate ways to achieve provable security against the emerging differential and linear attacks, using a small number of rounds (smaller than, say, DES). Few years later, such proposal was broken by a new technique called interpolation attack [JK97], and basically it was never re-considered – a recent standard textbook [KR11, Sect. 8.4] explicitly considers such constructions as "*not serious, for various reasons*".

We pick up this work from almost 20 years ago, and study if a simplified version of $\mathcal{PURE}$ with a much higher number of rounds can make this design secure. The cubic mapping is used as the

[2] The name MiMC is due to the goal of "Minimize the Multiplicative Complexity".

main component there and is also the main component of MiMC. It turns out, perhaps surprisingly, that the required much higher number of rounds (in the order of 100s instead of 10 or less) is very competitive when it comes to the new application areas of symmetric cryptography that motivate this work. MiMC - which can be instantiated both in $GF(p)$ and in $GF(2^n)$ - can be used for encryption as well as for collision-resistant cryptographic hashing based on a sponge construction.

One drawback of MiMC is that the decryption process is much more expensive than the encryption one. Moreover, MiMC does not outperform LowMC in PQ-Signature schemes applications. For this reason, we started considering possible variants of MiMC.

In Chapter 11, we present "Generalized Feistel MiMC" [AGP+18] (or GMiMC for simplicity), a first variant of MiMC obtained by simply turning the MiMC Even-Mansour cipher into a Feistel one, for which the encryption process and the decryption one are identical expect for the order of the round keys and round constants. On the other hand, one inconvenience of such a design is the possibility to set up competitive Meet-in-the-Middle attacks, which requires (approximately) to double the number of rounds with respect to MiMC in order to guarantee the same security. As a result, it seems that Feistel MiMC can not be competitive for the applications that we have in mind (where the goal is to minimize the number of multiplications), and that the only advantage of the Feistel approach seems to be that decryption is as cheap as an encryption computation. If this is true for Feistel MiMC, our current analysis suggest that this conclusion does not hold for Generalized Feistel constructions [Nyb96]. In particular, our analysis suggests that for unbalanced Feistel schemes with an expanding round function we do not have to increase the number of rounds further for $t > 2$ branches – up to a certain *finite* limit $t \leq t^\star$ – compared to $t = 2$ branches. From the practical point of view, GMiMC improves the performance of MiMC in several applications, like PQ-signature schemes, MPC and ZK protocols.

In Chapter 12, we propose another possible generalization of MiMC called HadesMiMC [GLR+19], constructed using our new Hades strategy. Hades strategy is a high-level design approach for cryptographic permutations and keyed permutations addressing needs of new applications that emphasize the role of multiplications in such designs, with a focus on simple arguments for its security. It builds up on the Wide-Trail design strategy for SP-Networks, which proved already very useful for a plethora of cipher and permutation designs as it helps to argue security against important classes of cryptanalytic attacks such as differential or linear attacks in a clean and simple way. Our approach "Hades" additionally allows for such arguments against important classes of algebraic attacks that are of much more concern when multiplications are to be minimized in a design. An important reason why this approach simultaneously enjoys elegant arguments against a larger number of classes of attacks and at the same time results in the most competitive instantiations to date is that we use a freedom in the design space that was so far not exploited: moving from an even to a highly uneven distribution of non-linearity, and hence cryptographic strength, of the rounds.

For our concrete instantiations HadesMiMC for the PQ digital signature scheme and MPC use-cases, we borrow ideas from the pre-predecessor of Rijndael/AES, namely SHARK [RDP+96], an S-Box based design with a single large MDS layer covering the whole internal state.

Remark. *Since I did not work on the practical applications/implementations of MiMC, GMiMC and HadesMiMC, I limit myself to recall in this Thesis the main results and I refer to the corresponding papers for a detailed discussion on such topic.*

Publication List. This second part includes contributions published as part of following paper at Asiacrypt 2016 as:

- M. R. Albrecht, L. Grassi, C. Rechberger, A. Roy, and T. Tiessen. MiMC: Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity. In: Advances in Cryptology – ASIACRYPT 2016. Ed. by J. H. Cheon and T. Takagi. Vol. 10031. LNCS. Springer, 2016, pp. 191–219. DOI: 10.1007/978-3-662-53887-6_7.

- M. R. Albrecht, L. Grassi, L. Perrin, S. Ramacher, C. Rechberger, D. Rotaru, A. Roy, and M. Schofnegger. Feistel Structures for MPC, and more. In Submission. 2018.

- L. Grassi, R. Lueftenegger, S. Ramacher, C. Rechberger, D. Rotaru, and M. Schofnegger. On a Generalization of Substitution-Permutation Networks: The HADES Design Strategy. In Submission. 2019.

2 Preliminary

In this chapter, we introduce the concept of block cipher, cryptography permutation and hash function, discuss the different notions of security, and take a look at some basic design strategies.

Remark. *The way in which this chapter and part of the next one are presented is (largely) inspired by the introduction of the PhD Thesis of Tyge Tiessen [Tie16b] and of Maria Eichlseder [Eic18]. We emphasize that the only goal of this chapter is to recall (basic) concepts useful for the topics discussed in this thesis. For a complete introduction on symmetric cryptography, we recommend the textbooks [MVO96] and [KR11].*

2.1. Block Cipher

A block cipher is a family of bijective functions Enc_k parameterized by a key $k \in \mathfrak{K}$ (where $|\mathfrak{K}| < \infty$) that map a finite set of messages $\mathfrak{P}$ to a finite set of encrypted messages $\mathfrak{C}$:

$$Enc_k(\cdot) : \mathfrak{P} \mapsto \mathfrak{C}.$$

The input to a block cipher is called the plaintext, the output is called the ciphertext. These are usually denoted as p and c. Accordingly $\mathfrak{P}$ and $\mathfrak{C}$ are called the plaintext space (or message space) and the ciphertext space. $\mathfrak{K}$ is called the key space. The function Enc_k is called the encryption function, its inverse $Dec_k := Enc_k^{-1}$ is called the decryption function.

Usually $\mathfrak{P}$, $\mathfrak{C}$, and $\mathfrak{K}$ have a group structure e.g. $(\mathbb{F}_q)^t$ where $q = 2^n$ or $q = p$ prime. Since the most common choice is the binary representation $(\mathbb{F}_{2^n})^t$, their elements can be associated with bit-strings of length $n \cdot t$.

2.1.1. Security Notion

Roughly speaking, a cipher can be considered secure if an adversary is not able to deduce any information in a faster way than a brute force attack. In such a brute force attack (or exhaustive search attack), the attacker simply tries out all keys, until she recovers the right one.

Note that such attack is always possible *if* the attacker knows the details of the block cipher. To thwart this type of attack one could simply try to keep the block cipher itself a secret. While this has been done in the past, modern cryptosystem are designed following Kerkhoffs' Principle:

Kerkhoffs' Principle [Ker83]. *The security of a cipher should solely rely on the secrecy of the key and never on the secrecy of the encryption method.*

The motivation behind this principle - which is nowadays generally accepted - is very simple. If a cipher is public, cryptographers are free to study it and assess its quality. This allows us to discard bad algorithms and to increase our confidence in the good ones.

Attack Models. In this context, an attack against a cryptosystem is a method which recovers some information about the plaintext and/or about the key. For being able to attack the cipher, an attacker adversary needs access to some data. Depending on them, it is common to define four different main types of attack scenarios:

- *Ciphertext-only attacks:* The adversary is only given access to a number of ciphertexts.
- *Known-plaintext attacks:* The adversary is given access to a number of corresponding plaintext-ciphertext pairs.
- *Chosen-plaintext attacks:* The adversary can choose one set of plaintexts for which she will be given the corresponding ciphertexts.
- *Adaptively chosen-plaintext attacks:* The adversary has access to the encryption of any plaintext of her choice during the whole attack.

Clearly each attack type gives the adversary more power than the previous attack types. While especially the chosen-plaintexts attacks might seem to be giving the adversary unrealistically much access, there are practical scenarios in which attacks of this type are possible. Furthermore, as security against chosen-plaintext attacks is a strong security notion, a cipher that is secure in this model is also secure in the weaker ones.

Data and Computational Costs. Since block ciphers are finite objects, they can always be attacked just given enough time and data (e.g. it is for example always possible to find the key by exhaustive search). To make meaningful statements about the security of a block cipher, it is hence necessary to state what the maximally allowed time and the maximally allowed amount of data are. The natural upper bound for the allowed time - the so-called time complexity - is the time needed to exhaustively try out the whole key space, while the natural upper bound for the maximally allowed knowledge of data - the so-called data complexity - is the whole codebook, i.e. all possible plaintext-ciphertext pairs.

2.2. Block Cipher - Design

A block cipher can be seen as a set of 2^k permutations on n-bit words which are indexed by the key k. Ideally we want a block cipher to randomly draw 2^k permutations out of the possible $2^n!$ permutations on n-bit words. Unfortunately, in practice such a random block cipher would be very difficult to implement in an efficient way for any meaningful key and block sizes. Indeed, for a block cipher that works on n-bit words, one needs a key of size at least k where

$$2^k \geq 2^n! \approx \sqrt{2\pi \cdot 2^n} \cdot 2^{n \cdot 2^n} \cdot e^{-2^n} \quad \rightarrow \quad k \geq 2^n \cdot (n - \log_2 e) + \frac{1 + \log_2(\pi \cdot n)}{2},$$

that is $k = \mathcal{O}(2^n \cdot n)$. For used values of n like $64, 128$ or 256, this means respectively a key size of at least $2^{70}, 2^{135}$ or 2^{264} bits.

To overcome this problem, most block ciphers are constructed combining simple building blocks in order to get a complex function, keeping in mind that it must be difficult for an adversary to find any relationship between plaintext, ciphertext and the secret key faster than via a brute force attack. One of the first constructions of this kind - the *product cipher* - was described by Shannon in [Sha49]

> *An iterative block cipher is an algorithm which maps a plaintext of fixed size n into a fixed size ciphertext n using a key K, by repeatedly applying a round transformation $f^i(\cdot)$ to the plaintext*
>
> $$E_K(\cdot) = f^r_{k^r}(\cdot) \circ \ldots \circ f^1_{k^1}(\cdot).$$

> *The round keys $k_1, k_2, ..., k_r$ are derived from K by a so called key schedule. The intermediate outputs of the function are called intermediate states. If the round functions are all equal we also refer to this as an iterated block cipher.*

Even if any round transformation $f_k(\cdot)$ is potentially possible, the largely choice is

$$f_k(\cdot) = k \star \hat{f}(\cdot),$$

where $\hat{f}(\cdot)$ is a function (usually a permutation) which is independent of the key, and where $\star$ is an operation in the structural group $(\mathbb{F}_q)^t$.

Given such a construction, the job of a designer is to determine (*1st*) suitable round functions and (*2nd*) the number of rounds needed to achieve security. *Potentially, given any arbitrary non-linear function, the corresponding iterative cipher can be secure by choosing a "largely enough" number of rounds.* The resulting performance, however, is of course unacceptable for many applications. The art is to define a round function in such a way that the designer can claim with reasonably confidence that a relatively limited number of rounds will provide the expected security level.

Due to a lack of any method to ensure that an efficient cipher design is secure against all possible attacks, the best option of determining the number of rounds of a cipher is to ensure that the cipher is secure against all known attacks. Quoting an article from Claude Shannon [Sha49]:

> *It is not enough merely to be sure none of the standard methods of cryptanalysis work - we must be sure that no method whatever will break the system easily. This, in fact, has been the weakness of many systems; designed to resist all the known methods of solution, they later gave rise to new cryptanalytic techniques which rendered them vulnerable to analysis.*

2.2.1. Substitution-Permutation Networks and Feistel Construction

Substitution-Permutation Networks

One of the largest used construction for block cipher is Substitution Permutation Networks (SPNs).

Referring to the "product cipher" previously recalled, the round function $f(\cdot)$ is constructed by combining two different operations: after splitting the whole message in smaller parts, a non-linear layer on each part and a (cheap) linear mixing operator on the whole state. In more details:

Substitution Layer: in the substitution layer, the state is separated into smaller segments, usually all of the same size. Each of the segments is then substituted independently of the other segments according to a so-called S-Box ("S" as in substitution) which is simply a look-up table. Depending on the design, the same S-Box might be used for all segments, or different S-Boxes can be employed.

Permutation Layer: in the permutation layer operates on the entire state. As such it is desirable to keep the complexity of this layer low to achieve an efficient implementation. Generally, the permutation layer is defined by a linear or an affine transformation.

To mix a round key into the state, the most common technique is though to add a key of the same length as the state to the state using either modular or exclusive-or addition. In this case, the function $f_k(\cdot)$ can be rewritten as

$$f_k(x) = k \oplus P \circ S(x)$$

where an initial key addition is performed. A schematic representation of an SPN scheme is proposed in Fig. 2.1.

[1] ***Acknowledgement.*** *The source-code of Figure 2.1 – made by Jérémy Jean – has been copied from [Jea16b].*

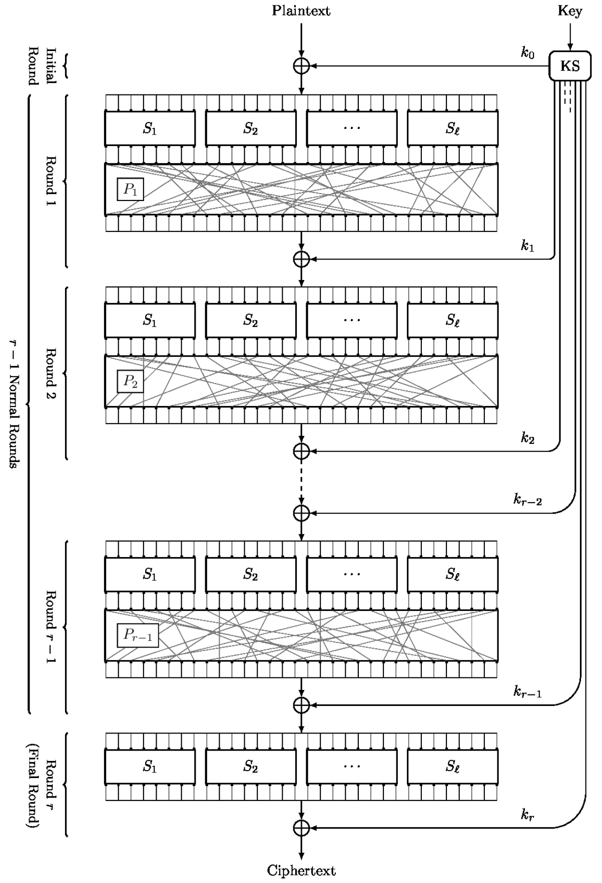


Figure 2.1.: A key-alternating Substitution-Permutation Networks (SPN)[1]

Feistel Construction

The basic idea underlying Feistel ciphers, also called Feistel networks, is similar to that of substitution-permutation networks. The main difference with substitution-permutation networks is that Feistel ciphers do not apply the complex transformations to all segments in parallel but apply the transformation only to a subset of segments each round.

In the classical Feistel construction, the state is split into two equally sized segments. The first segment is sent through some transformation, usually called the (Feistel) round function, and then mixed with the second segment via a commutative operation (e.g. a XOR-addition). The result of this operation together with the original value of the first segment become the two input segments of the next round - only with reversed roles. It follows that the function $f_k(\cdot)$ can be rewritten as

$$f_k\left(\left[x^L||x^R\right]\right) = \left[x^R||x^L \oplus F_k\left(x^R\right)\right]$$

where $x = \left[x^L||x^R\right]$ with $|x| = |x^L| + |x^R|$. See Fig. 2.2 for a schematic of a Feistel network.

SPN *versus* Feistel construction

Each one of the constructions has some advantages and some disadvantages. First of all, the round transformation of a Feistel scheme does not have to be invertible, giving the designer more options to choose from. Furthermore, the decryption algorithm can easily be derived by changing the order of the round keys. Also the computationally expensive round transformation are only applied to half

[2] ***Acknowledgement.*** *The source-code of Figure 2.2 – made by Jérémy Jean – has been copied from [Jea16b].*

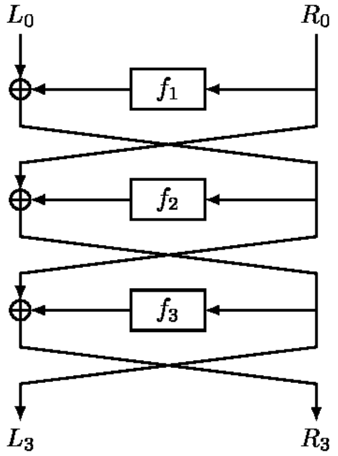


Figure 2.2.: A 3-round Feistel Network[2]

the state. But then, as only a part of the state undergoes a non-linear transformation at each round, generally more rounds are needed to achieve security than in comparable SPN constructions.

2.3. Block Cipher - Attack Scenario

To define the security of a cipher, we need to both define the power and the goal that the attacker tries to achieve. In the following, we briefly list a number of different possible attack goals:

Key-recovery attack: in a key-recovery attack, the attacker's goal is to retrieve the key. Since it is always possible to find key by exhaustive search, a key-recovery attack is considered to be meaningful if it has a time complexity below that of a brute-force attack[3].

Deduction attack: in a global deduction attack, the goal of the attacker is to find an efficient method for decrypting ciphertexts. Note that this is a weaker attack goal, since it is potentially possible to find such method without knowing anything about the key.

Distinguisher attack: in a distinguishing attack, the attacker is given access to both the cipher with a uniformly randomly chosen key and to a function that has been chosen uniformly at random from all invertible mappings from the plaintext space to the ciphertext space. The goal of the attacker is then to determine which of the two is the cipher and which is the random function.

As we are going to show in the following, this kind of attack is not only of theoretical interest: if the attacker can exploits a property which is independent of the secret key in order to set up a distinguisher attack, then such a property can potentially become the starting point of a key-recovery attack.

For completeness, we just mention that there exists a range of other types of adversaries and attack goals that give rise to other interesting security notions.

Ideally, a (block) cipher can be considered secure if it behaves like an *idealized block cipher*:

An ideal cipher Π *for a plaintext space* $\mathfrak{P}$*, a ciphertext space* $\mathfrak{C}$ *and a key space* $\mathfrak{K}$

$$\Pi(\cdot,\cdot) : \mathfrak{P} \times \mathfrak{K} \mapsto \mathfrak{C}$$

is a family of functions indexed by $k \in \mathfrak{K}$ *where each function is chosen independently and uniformly at random from all bijective* $\mathfrak{P}$ *to* $\mathfrak{C}$*.*

[3] However, it is principally possible for the designers of a cipher to explicitly state lower security claims, depending on the applications - e.g. a low-data application.

An ideal cipher tries to capture the intuitive notion of what we would like a cipher to behave like. It is important to note that the ideal cipher is not a block cipher, but it corresponds rather to the set of all possible block ciphers (for given $\mathfrak{P}$, $\mathfrak{C}$ and $\mathfrak{K}$) endowed with the uniform probability distribution. This means that no concrete instantiation can ever be an ideal cipher: the best we can hope for is thus that a good concrete design is indistinguishable from an ideal cipher.

2.3.1. Attack Construction

To construct an attack which can be executed faster than a brute-force attack, some weakness in the cipher design needs to be exploited. One of the widely used strategy to set up a key-recovery attack is by combining a "distinguishing attack" (which is independent of the secret key) and a "partial key-guessing".

A (secret-key) distinguisher is a property of the cipher that holds for many (potentially any) key and would be highly improbable to hold for a random permutation. Assume that the distinguishing property is determined from some of the input bits and some of the output bits of round s of the cipher. Obviously, such output bits of round s of the cipher can be written as a function of the output bits of rounds $s + r'$ for each $r' \geq 1$ and of the key bits after. Thus, let's assume that for a particular number of rounds $s + r$, such a function does not depend on all key bits. In such a case, the attacker can then determine the value of these key bits by using exhaustive search only on them. If this technique can successfully be applied, it allows to strongly reduce the time complexity to determine the full key.

In more details, an attacker given access to the output bits of round $r + s$ and the corresponding input bits but not the intermediate bits, can now try all key bit combinations needed to determine the intermediate state bits

$$\text{plaintexts} \xrightarrow[\text{distinguisher}]{R^s(\cdot)} \textit{“property”} \xleftarrow[\text{attack: key-guessing}]{R^{-r}(\cdot)} \text{ciphertexts.}$$

By discarding all those key bit combinations that do not give intermediate state bits for which the distinguishing property holds, the attacker can reduce the possible key space, potentially to the point of determining the key exactly.

In order to work, a crucial assumption – commonly known as the *wrong-key randomization hypothesis* – is needed: the property at round s must be unlikely to hold if the output of round $r + s$ is partially decrypted with a wrong key bit guess. To be more concrete, let $r = 1$ for simplicity, and let k and k' respectively the secret and the guessed key. Given a ciphertext $c = R^{1+s}(p)$, the attacker partial decrypt it using the guessed key, that is $R^{-1}(c \oplus k')$. Since $c = R(t) \oplus k = R(R^s(p)) \oplus k$, the distinguisher property holds with prob. 1 if and only if $k = k'$ (in this case: $R^{-1}(c \oplus k') = R^{-1}(R(t) \oplus k \oplus k') = t$). In all other cases $k \neq k'$, the text $R^{-1}(c \oplus k') = R^{-1}(R(t) \oplus k \oplus k')$ can potentially assume any possible value. Here the *wrong-key randomization hypothesis* plays a crucial role, since the key-recovery attacks work only if the distinguisher property does not hold (this allows the attacker to distinguisher the secret key from all other candidates).

2.3.2. "Academic" Attacks

From a research point of view, a cipher is considered broken if a weakness in the cipher that can be exploited with a complexity less than brute-force is found. Note that breaks might also require unrealistic amounts of known or chosen plaintext or unrealistic amounts of storage. Simply put, a break can just be a "certificated weakness": evidence that the cipher does not perform as advertised. In other words, in academic cryptography, a weakness or a break in a scheme is usually defined quite conservatively: it might require impractical amounts of time, memory, or plaintexts.

Academic attacks are often against weakened versions of a cryptosystem, such as a block cipher with some rounds removed. Almost all attacks become exponentially more difficult to execute as rounds are added to a cryptosystem, so it's possible for the full cryptosystem to be strong even though reduced-round variants are weak.

Attacks on "round-reduced" ciphers are important for several reasons. First of all, they enable us to assess the remaining *security margin* of that cipher, defined by the ratio between the number of rounds which can be successfully attacked and the number of rounds in the full cipher. More formally, given a cipher with n rounds, if there exists a cryptanalysis attack against a reduced-round version with $n - k$ rounds, the cipher has an absolute security margin of k rounds, or a relative security margin of k/n. The security margin provides a roughly estimation of the resistance of the cipher against cryptanalysis. However, note that it says nothing about the likelihood of these advances in cryptanalysis or about the resistance of the cipher against unknown attacks.

As second reason, attacks on round-reduced cipher enable us to develop new attack techniques which may become increasingly potent with additional improvements. Finally, it is possible that reduced round of particular ciphers (e.g. AES) have nice and well-studied properties that can be favorably as components of larger designs/schemes. As a result, successful cryptanalysis of these variants can be used to attack those schemes.

2.4. Design Challenges

Designing an efficient cipher is an open challenge. Depending on the particular application of a cipher, several trade-off must be taken in consideration.

If the goal is to design a cipher which can be efficiently implemented both in software and in hardware, traditionally it is a well accepted practice to have approximately the same number of linear and non-linear building blocks[4]. On the other hand, this conclusion does not hold for other applications, which include

- Lightweight Cryptography
- Specialized (block) Ciphers for Efficient Masking
- *Specialized ciphers for Multi-Party Computation (MPC), Fully-Homomorphic Encryption (FHE), Zero-Knowledge Proof (ZK), Post-Quantum Signature Schemes, ...*

among others. Roughly speaking, in all these cases non-linear operations are usually much more expensive than the linear ones. In the following, we limit ourselves to focus on the third application, which is the only one targeting in this thesis.

Specialized Ciphers for Multi-Party Computation (MPC), Fully Homomorphic Encryption (FHE), Zero-Knowledge Proof (ZK) and Post-Quantum Signature Schemes

Modern cryptography developed many techniques that go well beyond solving traditional confidentiality and authenticity problems in two-party communication. Examples of these are secure multiparty computation (MPC), zero-knowledge proofs (ZK), fully homomorphic encryption (FHE) and Post-Quantum (PQ) Signature Schemes.

While linear and non-linear building blocks have roughly similar costs in hardware and software implementations, the situation is radically different when a block cipher is used in applications like MPC, FHE, ZK and PQ signature schemes. In these cases, linear operations come almost for free, whereas nonlinear operations that involve symmetric cryptographic operations and communication between parties are very expenses. As a result, in the context of MPC, ZK, or FHE schemes, the goal becomes to design ciphers which minimize the number of non-linear operations.

[4] Linear and non-linear building blocks have roughly similar costs in hardware and software implementations. In CMOS hardware, the smallest linear gate (XOR) is about 2-3 times larger than the smallest non-linear gate (typically, NAND).

Fully-Homomorphic Encryption

The purpose of homomorphic encryption is to allow computation on encrypted data. This means that if a user has a function f and want to compute $f(m_1, ..., m_n)$ for some inputs $m_1, ..., m_n$, he can compute an equivalent function f' on encryptions of these inputs, $c_1, ..., c_n$, obtaining a result which decrypts to $f(m_1, ..., m_n)$

$$f(m_1, ..., m_n) \equiv Dec\left(f'\left(Enc(m_1), ..., Enc(m_n)\right)\right) .$$

To practically set up this process, in typical applications of homomorphic encryption, a plaintext p is encrypted under a public key pk and the corresponding ciphertext $c = HE_{pk}(m)$ is sent to some third-party evaluator. All known fully/somewhat homomorphic encryption schemes come with significant, often prohibitive ciphertext expansion. To prevent the thousand-fold to million-fold ciphertext expansion in (F)HE schemes, a form of compression is achieved using hybrid encryption. Given a symmetric encryption scheme E, a random key k is picked and a much smaller ciphertext

$$c = (HE_{pk}(k), E_k(m))$$

is sent. The third party then decompresses homomorphically into the original $c = HE_{pk}(m)$ using a particular decryption circuit. One downside of this approach is that application-specific operations on the ciphertext become more costly, as the decryption circuit of the cipher always needs to be evaluated as well.

Currently, all known candidates for FHE schemes use noise-based cryptography. Each operation on the homomorphically encrypted ciphertext incurs an increase in the noise. To prevent bootstrapping (necessary to reduce the noise), one needs to choose the FHE parameters generously enough to accommodate all additional noise from the decryption circuit. This is linked to the homomorphic capacity of a concrete instantiation of an FHE scheme, i.e. the number of operations on the ciphertext before an expensive bootstrapping operation is needed.

In many schemes, the noise level grows fast with the multiplicative depth of the circuit [BGV12; CLT14], that is in all somewhat and fully homomorphic encryption schemes known so far XOR (resp. addition) gates are considerably cheaper than AND (resp. multiplication) gates. Hence, symmetric encryption scheme proposals aiming for these types of applications minimize first of all the ANDdepth. Moreover, XOR gates do not increase the noise much, whereas AND gates increase the noise considerably [HS14]. Hence, as in somewhat homomorphic encryption schemes the parameters must be chosen such that the noise of the result is low enough to permit decryption, the overall complexity depends on the ANDdepth. Finally, while the cost of the application-specific homomorphic operations only depends on the ANDdepth of the cipher, the cost of evaluating the additional decryption circuit itself primarily depends on the number of multiplications.

In conclusion, both the ANDdepth and the number of AND computations are the most relevant metrics for this application.

Applications of Secure Multi-Party Computation Protocols

Secure multi-party computation (MPC) allows a set of parties to jointly evaluate a function on private inputs, with the guarantee that no party can learn anything more than the output of the function. In the last decade, MPC has moved from a theoretical pursuit to a very practical field, as protocols have become more efficient and many implementations been been developed.

For many years now, the *de facto* benchmark for MPC systems has been secure computation of the AES function [PSSW09; DK10; DKL+12]. Although the actual choice of this function was originally as a test-bed for comparing protocols, it has often been justified as “useful”, e.g. in the case in which an application needs to evaluate a symmetric encryption scheme with a secret-shared key. However, if this is indeed required, then there is no particular reason why AES should be the best choice to work with MPC, compared with other PRFs or symmetric ciphers.

There are various classes of practically efficient secure multiparty computation (MPC) protocols for securely evaluating Boolean circuits where XOR gates are considerably cheaper (no communication, less computation) than AND gates, e.g. protocols based on Yao's garbled circuits [Yao86]. These MPC protocols have a constant number of rounds – roughly speaking, the complexity of each round is linear in the ANDdepth of the evaluated circuit, and their total amount of communication depends on the "Multiplicative Complexity" of the circuit (each AND gate requires communication).

In conclusion, both the ANDdepth and the number of AND computations are the most relevant metrics for these MPC protocols.

Zero-Knowledge Proofs

Zero-knowledge proofs [GMR85; GMR89] are protocols that enable a prover to convince a verifier about the truth of a statement without leaking any information but the fact that the statement is true. In the case of non-interactive zero-knowledge (NIZK) proofs introduced by Blum, Feldman and Micali in [BFM88], the prover outputs just *one* message called a proof, which convinces the verifier of the truth of the statement. The central properties of (non-interactive) zero-knowledge proofs are completeness, soundness and zero-knowledge.

In several zero-knowledge proof protocols XOR relations can be proven for free and the complexity essentially depends on the number of AND gates of the relation to be proven. Examples for such protocols are presented in [BC86; BDP00] and more recently in [JKO13], where only one evaluation of a garbled circuit [Yao86] is required and that can make use of the free XOR technique [KS08].

SNARKs. A special kind of *Succinct Non-interactive Argument of Knowledge* [BCG+13] - or SNARK - was proposed in 2014 to build Zerocash [BCG+14], a digital currency similar to Bitcoin but achieving anonymity. A zk-SNARK is a non-interactive zero-knowledge proof of knowledge that is succinct, i.e. for which proofs are very short and easy to verify. The main idea of the SNARK is to provide a circuit whose satisfiability enables a verifier to check correctness of an underlying computation.

ZKBoo. ZKBoo [GMO16] is a proposal for practically efficient zero-knowledge arguments especially adapted for Boolean (e.g. arithmetic and binary) circuits. It is based on the "MPC-in-the-head" approach to zero-knowledge of Ishai *et al.* [IKOS09] (IKOS)[5]. The possibility to use this strategy to construct ZK protocols with good asymptotic properties has been showed in [IKOS09] for the first time, while in [GMO16] authors show how to exploit it to set up practically efficient ZK protocols.

A recent improvement in this topic includes Ligaro [AHIV17], a simple zero-knowledge argument protocol whose communication complexity is proportional to the square-root of the verification circuit size. Relevant for this thesis, this protocol can be based on any collision-resistant hash function. For completeness, we mention that both ZKBoo and Ligaro can be instantiated with MPC protocols in the preprocessing model, which allows shorter proofs (see [KKW18] for details).

Post-Quantum Signature Scheme

A possible way to construct a post-quantum signature scheme without relying on structured hardness assumptions involves symmetric key primitives. This is due to the fact that symmetric key primitives are conjectured to remain secure in the advent of sufficiently powerful quantum computers, while it is a well known fact that such quantum computers would break all discrete log and RSA based public key cryptosystems [Sho99].

Fish and Picnic [CDG+17] are new classes of digital signature schemes which derive their security entirely from the security of symmetric-key primitives, have extremely small key pairs, and are highly

[5] Since the details of such approach are not used in the following, we refer to [IKOS09] for more information.

parameterizable. The construction is based on a one-way function $f(\cdot)$, where for the secret key x, the image $y = f(x)$ is published as the public key. A signature on a message is then obtained from a non-interactive zero-knowledge proof of the relation $y = f(x)$, that incorporates the message in the challenge generation. This proof uses an improved ZKBoo [GMO16] – called ZKB++ – and the Fiat-Shamir transform [FS86] or the Unruh transform [Unr15] to make the zero-knowledge proof non-interactive.

Privacy-Preserving and Functional Signatures based on Symmetric-Key Primitives. Continuing the work on signatures, authors in [DRS18b] present a construction of a *ring signature* scheme solely relying on symmetric-key primitives. There the statement is extended to prove an authentication path in a Merkle tree in zero-knowledge, hence collision-resistant hash functions with low multiplicative complexity are of interest for this application.

Besides [DRS18b], ring signature schemes are also considered in [KKW18] and in [BEF18]. Here authors construct group signature schemes built only from symmetric primitives, such as hash functions and PRFs, widely regarded as the safest primitives for post-quantum security. Finally, we mention [DRS18a], where authors present a double-authentication preventing signatures (DAPS) based on symmetric primitives. DAPS are a variant of digital signatures used to sign messages of the form $m = (a, p)$ with a being the so called address and p the payload.

2.5. Brief Introduction to (Cryptography) Permutations

A *cryptographic permutation* P is a bijective function

$$P : \mathfrak{T} \to \mathfrak{T}$$

(usually $\mathfrak{T} \equiv (F_q)^t$ for $q = 2^n$ or $q = p$ prime) such that $P(\cdot)$ (and, if required, its inverse P^{-1}) is easy to evaluate.

Due to the lack of a key, it is not easy to define a clear security notion for unkeyed cryptographic permutations. *To be considered secure for cryptographic applications, P must not permit any structural distinguisher.* This includes any property which is not expected for a randomly chosen permutation.

Probably, the simplest way to construct a cryptography permutation is to consider a block cipher with a fixed (random) key $\hat{K}$

$$P(\cdot) \equiv Enc_{\hat{K}}(\cdot).$$

In such a case, $P(\cdot)$ can be considered secure if and only if $Enc_{\hat{K}}(\cdot)$ is a secure cipher in the secret-/known-/chosen-key model, as described in detail in the following.

2.6. Brief Introduction to Hash Functions

A cryptographic *hash function* H is an efficient deterministic algorithm, which maps messages of arbitrary length to strings of fixed length

$$H : \mathfrak{M} \mapsto \mathfrak{H},$$

e.g. $\mathfrak{M} \equiv \mathbb{F}_{2^N}$ where N "$\to$" ∞ and $\mathfrak{H} \equiv \mathbb{F}_{2^n}$ for a fixed n.

The output of a hash function is called the hash value, hash, digest or finger- print of a message. Every possible message is associated with such a value which can then be used as a short identifier or representative for this message. Cryptographic hash functions are used to provide integrity and authenticity in a large number of applications and protocols.

2.6.1. Secure Hash Function

For a secure cryptographic hash function it should be difficult to find a message for a given hash value and it should also be difficult to find two messages which result in the same hash value, a *collision*[6]. More formally, a secure hash function must satisfy the following requirements:

Preimage Resistance: For a given output y it should be computationally infeasible to find an input x such that $y = H(x)$.

Second Preimage Resistance: For a given x and $y = H(x)$ it should be computationally infeasible to find $x' \neq x$ such that $H(x') = y$.

Collision Resistance: It should be computationally infeasible to find two distinct inputs x, x' such that $H(x) = H(x')$.

Note that collision resistance implies second preimage resistance.

An ideal hash function should behave like a *random oracle*, that is a function which outputs a random value for each new input. If an input value is repeated it outputs the previously used value. No practical hash function can implement a random oracle, as the description would be too large. Nonetheless, a good hash function should be difficult to distinguish from such a random oracle.

Generic Attacks. Similar to block ciphers, there exist also generic attacks on hash functions, which allow an attacker to find preimages or collisions disregarding the underlying structure. When treating the hash function as a black box the only relevant parameter for these attacks is the length of the hash value n.

An attacker can always find a (second) preimage by trying out many inputs and checking whether they give the desired hash value. The attack is likely to succeeds after trying approximately 2^n different inputs. Finding a collision for a hash function can be done more efficiently. Using the so called birthday paradox, a generic attack requires approximately $2^{n/2}$ different inputs to succeed.

2.6.2. Design – Sponge Construction

There are several ways to design a secure hash function, including the the Merkle-Damgård (MD) construction and the Sponge construction. For the goal of this thesis, we limit ourselves to recall the second one.

The sponge construction (Fig. 2.3) has been introduced by Bertoni, Daemen, Peeters and Van Assche [BDPA07; BDPA08] as a theoretical model for hash functions. Sponge construction is based on a wide random functions/permutations, and allows inputting ("absorbing" in sponge terminology) any amount of data, and outputting ("squeezing") any amount of data, while acting as a pseudorandom function with regard to all previous inputs. Keccak/SHA-3 [BDPA11] is based on such a construction.

A sponge construction is composed of two components:

- a state memory – denoted by S – containing n bits;
- a function $f(\cdot) : \{0,1\}^b \to \{0,1\}^b$ that transforms the state memory;
- a padding function P.

The state memory is divided into two sections: one of size r (the "bitrate") and the remaining part of size c (the "capacity"), such that $n = r + c$.

A sponge function operates as follows:

[6] As the input domain of a hash function is always significant larger than the output domain this collisions are unavoidable.

[7] ***Acknowledgement.*** *Figure 2.3 – made by Keccak Team – has been copied from* `https://keccak.team/figures.html`.

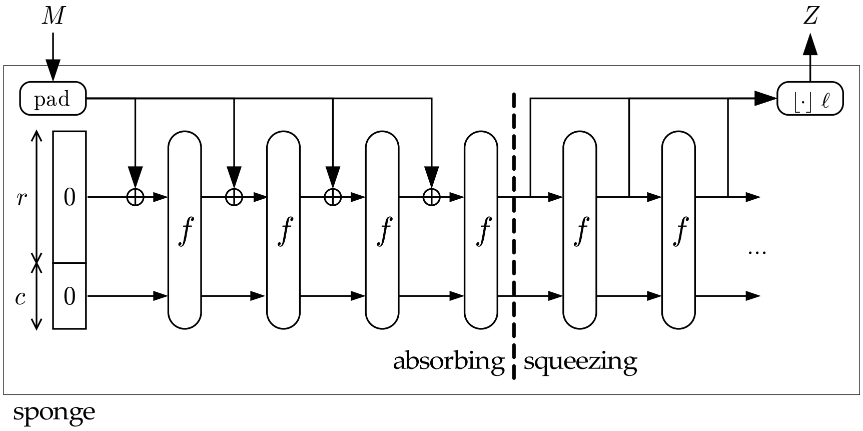


Figure 2.3.: A Sponge Hash Function[7]

Initialization: the state S is initialized to zero:

$$S_0 = (\underbrace{0, ..., 0}_{r \text{ bits}} \| \underbrace{0, ..., 0}_{c \text{ bits}}) \equiv (S^r \| S^c);$$

Absorbing: after transforming the input M into blocks of r bits (the initial message is padded using function P if necessary), that is $M = (M_1 \| ... \| M_m)$, the state is "absorbed" (in the sponge metaphor)

$$\forall i = 1, ..., m: \qquad S_i = P(S^r_{i-1} \oplus M_i \| S^c_{i-1});$$

Squeezing: the sponge function output $Z = (Z_1 \| Z_2 \| ...)$ is now ready to be produced ("squeezed out") as follows

$$\forall i \geq 1: \qquad S_{m+i} = P(S_{m+i-1}), \qquad Z_i = S^r_{m+i}.$$

As proved by Bertoni *et al.*[BDPA08], when the internal permutation $f(\cdot)$ is modeled as a randomly chosen permutation, the corresponding Sponge function is indifferentiable from a random oracle up to $2^{c/2}$ calls to $f(\cdot)$. This is due to the fact that distinguishing this Sponge construction from a random oracle requires the detection of *inner collisions* in the capacity part. As a result, a sponge with a capacity of c provides respectively $2^{c/2}$ collision and $2^{c/2}$ (second) preimage resistance.

2.7. Preliminary - Probabilistic Theory

Finally, we also recall some useful concepts regarding probabilistic theory.

Let $n \in \mathbb{N}$ and let $f(\cdot) : \mathbb{Z} \mapsto \mathbb{Z}$ be a discrete function. The n-th moment - denoted by μ_n - of the discrete function $f(\cdot)$ about a value $c \in \mathbb{R}$ is defined as

$$\mu_n = \sum_{x \in \mathbb{Z}} (x - c)^n \cdot f(x)$$

Similar definition can be given for real-valued continuous function $f(\cdot)$ of a real variable.

If $f(\cdot)$ is a discrete probability density function[8], then the value of the sum above is called the n-th moment of the probability distribution. In the following, we consider *the normalized n-th central*

[8] Remember that a discrete probability density function $f(\cdot) : \mathbb{Z} \mapsto \mathbb{Z}$ satisfies the following properties: *(1st)* $0 \leq f(x) \leq 1$ for each x and *(2nd)* $\sum_{x \in \mathbb{Z}} f(x) = 1$.

moment for $n \geq 3$, defined as the n-th central moment divided by $\left(\mu_2\right)^{n/2}$:

$$\mu_n = \frac{1}{\mu_2} \cdot \sum_{x \in \mathbb{Z}} (x - \mu_1)^n \cdot f(x) \equiv \frac{\sum_{x \in \mathbb{Z}} (x - \mu_1)^n \cdot f(x)}{\left(\sum_{x \in \mathbb{Z}} (x - \mu_1)^2 \cdot f(x)\right)^{n/2}} \qquad \forall n \geq 3,$$

where the constant c is defined as

$$c = \begin{cases} 0 & \text{if } n = 0 \\ \mu_1 & \text{otherwise} \end{cases}$$

Moreover, in the following we mainly focus on the first three central moment:

- the first (raw) moment of a random variable X is the *mean*, usually denoted by $\mu = \mathbb{E}[X]$;
- the second central moment is the *variance*, usually denoted by σ^2. The positive square root of the variance is the standard deviation $\sigma \equiv \left(\mathrm{E}\left[(x - \mu)^2\right]\right)^{\frac{1}{2}}$;
- the third central moment is the measure of the lopsidedness of the distribution; any symmetric distribution has a third central moment equal to zero. The normalized third central moment is called the *skewness*, often denoted by γ. A distribution that is skewed to the left (the tail of the distribution is longer on the left) will have a negative skewness. A distribution that is skewed to the right (the tail of the distribution is longer on the right), will have a positive skewness.

Part I.

Cryptanalysis of AES

3

Advanced Encryption Standard (AES)

In this chapter, we will first recall AES, and we give an overview of different types of attack elements and how they can be combined to form more complex attacks.

3.1. AES

The Advanced Encryption Standard (AES) [DR98; DR00; DR02b], also known by its original name Rijndael, is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001. AES is a subset of the Rijndael cipher developed by two Belgian cryptographers, Vincent Rijmen and Joan Daemen, who submitted a proposal to NIST during the AES selection process.

3.1.1. Description of AES

AES [DR02b] is a *Substitution-Permutation network* that supports key sizes of 128, 192 and 256 bits. The 128-bit plaintext initializes the internal state as a 4 × 4 matrix of bytes

$p_{0,0}$	$p_{0,1}$	$p_{0,2}$	$p_{0,3}$
$p_{1,0}$	$p_{1,1}$	$p_{1,2}$	$p_{1,3}$
$p_{2,0}$	$p_{2,1}$	$p_{2,2}$	$p_{2,3}$
$p_{3,0}$	$p_{3,1}$	$p_{3,2}$	$p_{3,3}$

as values in the finite field $\mathrm{GF}(2^8) \equiv \mathbb{F}_{256}$ defined using the irreducible polynomial $X^8+X^4+X^3+X+1$, that is $GF(2)[X]/(X^8+X^4+X^3+X+1)$. Depending on the version of AES, N_r rounds are applied to the state: $N_r = 10$ for AES-128, $N_r = 12$ for AES-192 and $N_r = 14$ for AES-256. An AES round applies four operations to the state matrix:

- *SubBytes* - S-Box$(\cdot)$;
- *ShiftRows* - $SR(\cdot)$;
- *MixColumns* - $MC(\cdot)$;
- *AddRoundKey* - $ARK(\cdot)$.

One round of AES can be described as

$$R(\cdot) = K \oplus MC \circ SR \circ \text{S-Box}(\cdot).$$

In the first round an additional AddRoundKey operation (using a whitening key) is applied, and in the last round the MixColumns operation is (usually) omitted[1].

[1] The choice to omit the final MixColumns operation allows to make the cipher and its inverse more similar in structure.

The SubBytes Step. In the SubBytes step, each byte $p_{i,j}$ in the state matrix is replaced with another one S-Box$(p_{i,j})$ using an 8-bit substitution box, the Rijndael S-Box

$$\forall i, j: \qquad p_{i,j} \mapsto \text{S-Box}(p_{i,j}).$$

This operation provides the non-linearity in the cipher. Only one S-Box is used for all bytes, which is derived from the multiplicative inverse over GF(2^8), known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-Box is constructed by combining the inverse function with an invertible affine transformation, that is S-Box$(x) = M \oplus x^{-1} \oplus 0\text{x}63$, where M is a $2^8 \times 2^8$ binary invertible matrix and $0^{-1} := 0$. The S-Box is also chosen to avoid any fixed point.

The ShiftRows Step. The ShiftRows step operates on the rows of the state. It cyclically shifts the bytes in each row by a certain offset. For AES, the r-th row is shifted of r to the left:

$$\forall i, j: \qquad p_{i,j} \mapsto p_{i,j+i \bmod 4}.$$

The importance of this step is to avoid that the columns are encrypted independently, in which case AES degenerates into four independent block ciphers. In other words, the aim is to guarantee that, given two bytes in the same column, they belong to different columns after this step.

The MixColumns Step. In the MixColumns step, the four bytes of each column of the state are combined using an invertible linear transformation. The MixColumns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes:

$$\forall j: \qquad \begin{bmatrix} p_{0,j} \\ p_{1,j} \\ p_{2,j} \\ p_{3,j} \end{bmatrix} \mapsto \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} p_{0,j} \\ p_{1,j} \\ p_{2,j} \\ p_{3,j} \end{bmatrix}$$

where each entry of the (fixed) matrix M is treated as element of GF$(2^8)[X]$. Together with ShiftRows, MixColumns provides diffusion in the cipher.

The AddRoundKey Step. In the AddRoundKey step, the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael's key schedule (each subkey has the same size of the state). The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR:

$$\forall i, j: \qquad p_{i,j} \mapsto p_{i,j} \oplus K_{i,j}.$$

3.1.2. Key-Schedule

Key-Schedule for AES-128. The key schedule of AES-128 takes the user key and transforms it into 11 subkeys of 128 bits each. The subkey array is denoted by $W[0, \ldots, 43]$, where each word of $W[\cdot]$ consists of 4 bytes and where the first 4 words of $W[\cdot]$ are loaded with the user secret key. The remaining words of $W[\cdot]$ are updated according to the following rule:

$$W[i][j] = \begin{cases} W[i][j-4] \oplus \text{S-Box}(W[i+1][j-1]) \oplus R[i][j/4] & \text{if } j \bmod 4 = 0 \\ W[i][j-1] \oplus W[i][j-4] & \text{otherwise} \end{cases}$$

where $i = 0, 1, 2, 3$, $j = 4, \ldots, 43$ and $R[\cdot]$ is an array of predetermined constants[3].

[2] ***Acknowledgement.*** *The source-code of Figure 3.1 – made by Jérémy Jean – has been copied from [Jea16b].*

[3] The round constants are defined in $GF(2^8)[X]$ as $R[0][1] = X$, $R[0][r] = X \cdot R[0][r-1]$ if $r \geq 2$ and $R[i][\cdot] = 0$ if $i \neq 0$. In the following, let $R[r] \equiv R[0][r]$.

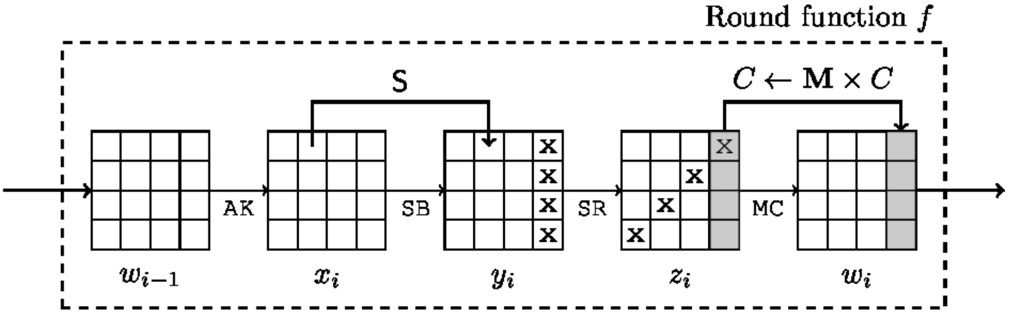


Figure 3.1.: Essential structure of an AES round[2]

Key-Schedule for AES-192. The key schedule of AES-192 is similar to the one given for AES-128. In this case, the subkey array is denoted by $W[0,\ldots,51]$, where here the first 6 words of $W[\cdot]$ are loaded with the user secret key. The remaining words of $W[\cdot]$ are updated according to the following rule:

$$W[i][j] = \begin{cases} W[i][j-6] \oplus \text{S-Box}(W[i+1][j-1]) \oplus R[i][j/6] & \text{if } j \bmod 6 = 0 \\ W[i][j-1] \oplus W[i][j-6] & \text{otherwise} \end{cases}$$

where $i = 0,1,2,3$, $j = 6,\ldots,51$ and $R[\cdot]$ is an array of predetermined constants.

Key-Schedule for AES-256. The case AES-256 is a little different from the previous cases. In this case, the subkey array is denoted by $W[0,\ldots,59]$, where here the first 8 words of $W[\cdot]$ are loaded with the user secret key. The remaining words of $W[\cdot]$ are updated according to the following rule:

$$W[i][j] = \begin{cases} W[i][j-8] \oplus \text{S-Box}(W[i+1][j-1]) \oplus R[i][j/8] & \text{if } j \bmod 8 = 0 \\ W[i][j-8] \oplus \text{S-Box}(W[i][j-1]) & \text{if } j \bmod 8 = 4 \\ W[i][j-1] \oplus W[i][j-8] & \text{otherwise} \end{cases}$$

where $i = 0,1,2,3$, $j = 8,\ldots,59$ and $R[\cdot]$ is an array of predetermined constants.

3.2. The Wide Trail Strategy

The strategy that has been used in the design of Rijndael, the block cipher which has been selected to become the Advanced Encryption Standard (AES), is the “Wide Trail” strategy. This is an approach for designing the round transformations of block ciphers that combine efficiency and resistance against differential and linear cryptanalysis. In order to explain the wide trail strategy, we first recall differential and linear cryptanalysis.

3.2.1. Differential Cryptanalysis

Differential cryptanalysis is one of the most effective and most versatile cryptanalytic techniques for analyzing symmetric primitives. The approach was introduced by Biham and Shamir [BS90; BS91; BS93], who exploited this technique in order to present the first attack faster than brute-force for full-round DES [BS92]. After this pioneer and original work, many variants of such strategy have been proposed in the literature. As a result, one of the main security requirements for any new design is security against differential cryptanalysis.

Roughly speaking, given pairs of inputs with some fixed input difference, differential cryptanalysis considers the probabilistic distribution of the corresponding output difference produced by the (round-reduced) cryptographic primitive. If such probabilistic distribution differs from a uniform distribution one, then the attacker can exploit such fact to set up an attack on the (round-reduced) cipher.

Basic Concepts of Differential Cryptanalysis. As largely done in the literature, we assume that the plaintexts space $\mathfrak{P}$ and the ciphertexts space $\mathfrak{C}$ are equal to a Galois Field $\mathfrak{P} \equiv \mathfrak{C} \equiv (\mathbb{F}_2^n, \oplus)$.

In differential cryptanalysis, one considers a pair of texts x and y, and evaluates their difference[4] $\Delta = x \oplus y$. Given an input difference Δ_I, the idea is to study the *probabilistic distribution* of the corresponding output difference Δ_O after a certain number of rounds r. Note that, since each round $f(\cdot)$ is composed by linear and by non-linear operations, a single input difference is usually mapped in many different output differences, that is $f(x) \oplus f(x \oplus \Delta_I)$ is in general different from $f(y) \oplus f(y \oplus \Delta_I)$ for $x \neq y$. Thus, in a natural way, one can define the probability of a *differential* $\alpha \to \beta$ as

$$Prob\big[\alpha \to \beta\big] = \frac{\big|\{x \in \mathbb{F}_2^n | f(x \oplus \alpha) \oplus f(x) = \beta\}\big|}{|\mathbb{F}_2^n|} = \frac{\big|\{x \in \mathbb{F}_2^n | f(x \oplus \alpha) \oplus f(x) = \beta\}\big|}{2^n}$$

where $|\cdot|$ denotes the cardinality of the corresponding set. In the following, we limit ourselves to recall that the maximum probability that $f(\cdot)$ maps an input difference α to an output difference β for uniformly random x is defined as the *maximum differential probability*, denoted by

$$DP_{max}(f) := \max_{\alpha \neq 0, \beta} \frac{\big|\{x \in \mathbb{F}_2^n | f(x \oplus \alpha) \oplus f(x) = \beta\}\big|}{|\mathbb{F}_2^n|}$$

Since $x \in \mathbb{F}_2^n$ satisfies $f(x \oplus \alpha) \oplus f(x) = \beta$ if and only if $x \oplus \alpha$ satisfies it, the maximum differential probability is always greater or equal to $2/|\mathbb{F}_2^n|$. Functions with $DP_{max}(f) = 2/|\mathbb{F}_2^n|$ are called *almost perfect nonlinear* (APN) and have been proposed for designing ciphers resistant against differential cryptanalysis [NK92; NK95].

Before going on, we recall that that $DP_{max}(f) = DP_{max}(f^{-1})$, that is a function f and its inverse f^{-1} have the same DP_{max} (see e.g. [Nyb94] for details).

Classical Differential Cryptanalysis

For a random function - and similarly for a random permutation, the probability of any given differential with non-zero input difference is very low, on average $1/|\mathbb{F}_2^n|$. If for a cipher $E(\cdot)$ there exists a differential of probability significantly different from $1/|\mathbb{F}_2^n|$, then an attacker can potentially use it to distinguish the cipher from a random function (or permutation), or to set up a key-recovery attack.

Even if it is theoretically possible to compute the probability of any differential for a product cipher $E(\cdot)$, from a practical points of view (at least) two problems arise:

1. How to determine the differential probability when exhaustively trying all text pairs is computationally infeasible?
2. How to determine the differential probability of functions with secret parameters (e.g. block ciphers with secret keys)?

Differential Characteristic. For simplicity, let us consider only the case of an iterative cipher $E : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$:

$$E(\cdot) = R_r \circ ... \circ R_2 \circ R_1(\cdot),$$

where the commutative operation defined on it is simply the XOR-sum $\oplus$. In this case, it is natural to consider intermediate differences taken between the single rounds:

[4] If one works on a different field/group, then the difference must be evaluated w.r.t. the operation that defines such field/group.

An $r+1$-tuple of differences $(\alpha_0, \alpha_1, ..., \alpha_r)$ with differences in $\mathbb{F}_2^n$

$$\alpha_0 \xrightarrow{R_1(\cdot)} \alpha_1 \xrightarrow{R_2(\cdot)} ... \xrightarrow{R_r(\cdot)} \alpha_r$$

is called a differential characteristic.

As we just mentioned, computing the probability of a single differential trail is in general *infeasible* for real ciphers.

The common way to compute such probability is to consider the product of the single round transition. The first problem that one encounters in such a task is that (*1st*) the probability of such differential characteristic depends on the concrete values of the initial pair of texts (that is, x and $x \oplus \alpha_0$) and that (*2nd*) the differences α_i are in general not independent, since the round functions $R_i(\cdot)$ are in general not independent. In other words, the probability of a single differential trail $(\alpha_0, \alpha_1, ..., \alpha_r)$ is given by

$$Prob_{x,K}\big[(\alpha_0, \alpha_1, ..., \alpha_r)\big] = \prod_{i=1}^{r} Prob_K\big[\alpha_i \,|\, \alpha_{i-1}, ..., \alpha_0, x\big],$$

where K denotes the secret key. While the assumptions of round independence and of independence of the initial value x are clearly not satisfied in general, experiments suggest that the approximation

$$\prod_{i=1}^{r} Prob_K\big[\alpha_i \,|\, \alpha_{i-1}, ..., \alpha_0, x\big] \simeq \prod_{i=1}^{r} Prob_K\big[\alpha_i \,|\, \alpha_{i-1}\big]$$

holds quite well in many practical cases. More formally, this corresponds to the assumption of “Markov ciphers” [LMM91] and of “independence of the rounds”. A common approach is to consider these two assumptions, and to assume that this model is close to reality.

A second problem that one encounters in such a task is that the round function $R(\cdot)$ depends on keys which are in general secret. To handle this problem and similar to before, the idea is simply to assume that the probability of a differential characteristic is independent of the value of the secret key. In other words, this assumption – known as “stochastic equivalence hypothesis” – states that the probability of a differential characteristic behave (almost) in the same way for all keys.

Using these two assumptions, the probability of a differential characteristic can be computed using the formula:

$$Prob_{x,K}\big[(\alpha_0, \alpha_1, ..., \alpha_r)\big] \simeq \prod_{i=1}^{r} Prob_{\hat{x},\hat{K}}\big[\alpha_{i-1} \xrightarrow{R_i(\cdot)} \alpha_i\big],$$

where $\hat{x}$ and $\hat{K}$ are randomly fixed values. For simplicity, in the following let

$$Prob_{\hat{x},\hat{K}}\big[\alpha_{i-1} \xrightarrow{R_i(\cdot)} \alpha_i\big] \equiv Prob\big[\alpha_{i-1} \xrightarrow{R_i(\cdot)} \alpha_i\big].$$

From Differential Characteristic to Differential Trail. In most attacks, the attacker has no information about intermediate states of the cipher, so it is in general infeasible to exploit the previous formula. Therefore for an attack we are actually interested in the probability of the *differential trail*

$$\alpha_0 \xrightarrow{E(\cdot)} \alpha_r : \qquad \alpha_0 \xrightarrow{R_1(\cdot)} ? \xrightarrow{R_2(\cdot)} ... \xrightarrow{R_{r-1}(\cdot)} ? \xrightarrow{R_r(\cdot)} \alpha_r$$

where the intermediate differences are not fixed. Thus, the probability of a differential (α_0, α_r) over an encryption function $E : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$ (where $E(\cdot) = R_r \circ ... \circ R_2 \circ R_1(\cdot)$) is the sum of the probabilities of all differential trails $(\alpha_0, \alpha_1, ..., \alpha_r)$ that it contains:

$$Prob\big[\alpha_0 \xrightarrow{E(\cdot)} \alpha_r\big] = \sum_{\alpha_1,...,\alpha_{r-1}} Prob\big[(\alpha_0, \alpha_1, ..., \alpha_r)\big] \simeq \sum_{\alpha_1,...,\alpha_{r-1}} \prod_{i=1}^{r} Prob\big[\alpha_{i-1} \xrightarrow{R_i(\cdot)} \alpha_i\big]$$

Obviously

$$Prob\big[(\alpha_0, \alpha_1, ..., \alpha_r)\big] \leq Prob\big[\alpha_0 \xrightarrow{E(\cdot)} \alpha_r\big],$$

which means that a differential characteristic allows us to determine a lower bound on the probability of a differential trail.

Key-Recovery Attack. *If there now exists a differential $\alpha_0 \to \alpha_r$ over r rounds of a n-bit cipher that has a probability p (much) larger than 2^{-n}, we can use this differential to distinguish these rounds from a random permutation in a chosen-plaintext attack.*

Moreover, this distinguisher can be used to set up a key-recovery attack on $r + s$ rounds. In more details, given a set of sufficiently many pairs of plaintexts $(p_i, p'_i \equiv p_i \oplus \alpha_0)$ for $i = 0, ..., N$, the attacker asks for the corresponding ciphertexts (c_i, c'_i) after $r + s$ rounds, guesses the last s sub-keys, partially decrypts s rounds and compute the probabilistic distribution of the difference at round r:

$$(p_i, p'_i \equiv p_i \oplus \alpha_0) \xrightarrow[\text{diff. Distinguisher}]{R^r(\cdot)} \text{probability } p \text{ that "}R^r(p_i) \oplus R^r(p'_i) = \alpha_r\text{"} \; > 2^{-n} \xleftarrow[\text{key-guessing}]{R^{-s}(\cdot)} (c_i, c'_i).$$

Obviously, the probability of α_r is equal to p for the right key guessing. Due to the "wrong-key randomization hypothesis" – which states that when decrypting one or several rounds with a wrong key guess creates a function that behaves like a random function, the attacker can be expect that the probability of α_r is approximately equal to 2^{-n} for a wrongly key guessing.

The data complexity primarily depends on the number of queries necessary s.t. at least one of them satisfies the given differential $\alpha_0 \to \alpha_r$ with non-negligible probability. Roughly speaking, this is given by the inverse probability $C \cdot p^{-1}$ (for some constant $C \geq 1$), but several additional parameters must also be taken into account. A detailed analysis of the success probability and its dependency on the invested data complexity has been performed by Selçuk [SB02; Sel08].

3.2.2. Linear Cryptanalysis

Linear cryptanalysis is a known-plaintext attack in which the attacker exploits probabilistic *linear* - or (more generally) affine - *relations* between bits of the plaintext, of the ciphertext and of the key. It was introduced by Matsui [Mat93] as a theoretical attack on the Data Encryption Standard (DES), and later successfully used in the practical cryptanalysis of DES [Mat94].

Let $\langle a, b\rangle$ denote the canonical inner product in $\mathbb{F}_2^n$

$$\langle a, b\rangle := \bigoplus_i a_i \cdot b_i.$$

A *linear approximation* of a cipher $E_K(\cdot)$ through masks α, β and γ is defined as

$$\langle \alpha, x\rangle \oplus \langle \beta, E_K(x)\rangle = \langle \gamma, K\rangle$$

where x and K denotes respectively a plaintext and the (secret) key. The *linear deviation* of such linear expression in linear cryptanalysis is given by $|p - 1/2|$, where p is the probability that the previous equality holds.

Given "good" masks α, β and γ (that is masks for which the linear deviation is different from 0), the value in the l.h.s. of the previous equation for a large number of plaintext/ciphertexts pairs can be exploited to derive information about the secret key by analyzing the value in the l.h.s. of the previous equation that occurs most often. In principle, this gives a single bit of information about the key. In [Mat93], it is shown that the probability of making a wrong guess is very small if the number of plaintext/ciphertexts pairs is larger than $|p - 1/2|^{-2}$.

Basic Concepts of Linear Cryptanalsis. For the follow-up, we recall some basic concepts of linear cryptanalysis. As for differential cryptanalysis, it is possible to assign to each function $f(\cdot)$ a *linear approximation table*, whose entry in row α and column β corresponds to

$$\big|\{x \in \mathbb{F}_2^n : \langle \alpha, x\rangle \oplus \langle \beta, f(x)\rangle = 0\}\,\big|.$$

Let the *linear correlation* of a function $f(\cdot)$ with masks α and β defined as[5]

$$\mathfrak{L}(f) := \max_{\alpha,\beta\neq 0} \mathfrak{L}_{\alpha,\beta}(f)$$

where

$$\mathfrak{L}_{\alpha,\beta}(f) := 2^{-n}\cdot \big|\{x \in \mathbb{F}_2^n : \langle \alpha, x\rangle \oplus \langle \beta, f(x)\rangle = 0\}\,\big|.$$

A function is an *Almost Bent* (AB) function if $\mathfrak{L}(f) = 2^{\frac{-n+1}{2}}$.

Theorem 1 ([CCZ98])**.** *Any almost bent (AB) function is also almost perfect nonlinear (APN).*

From this theorem and from the results proposed in [CV94] (where it is showed that the probability of a differential can be expressed in terms of a sum of correlations of linear approximations), it follows that linear-resistant functions are also differential-resistant. On the other hand, resistance against differential cryptanalysis does not imply resistance against linear cryptanalysis.

As we are going to show in the next section, (almost) bent functions are the ones that oppose an optimum resistance to linear cryptanalysis, as almost perfect nonlinear functions are the ones that oppose an optimum resistance to differential cryptanalysis.

3.2.3. The Wide Trail Design Strategy

The *wide-trail strategy* by Daemen and Rijmen [DR01; DR02a] is a general approach for designing the round transformation of key-alternating block ciphers that combines efficiency and resistance against linear and differential cryptanalysis. For simplicity, in the following we limit ourselves to focus on differential cryptanalysis (analogous results can be derived for linear cryptanalysis).

The Wide Trail Strategy. Consider a round transformation for key-alternating SP constructions for $t \cdot n = N$-bit blocks, built as a sequence of two invertible steps:

1. a local non-linear transformation, that is a permutation consisting of a number of n-bit S-Boxes (t bundles or cells);
2. a linear mixing transformation λ providing high diffusion.

Moreover, as largely done in the literature, assume[6] that a cipher is secure against differential cryptanalysis if each characteristic has probability smaller than 2^{-N}. Since a linear transformation does not affect the probability of a differential, the probability of a differential characteristic depends only on:

[5] To be more precise, the linear correlation of a function $f(\cdot)$ is usually defined using the *Fourier Transform* [Car10] as

$$\begin{aligned} corr(f) &= 2^{-n}\cdot \max_{\alpha,\beta\neq 0} \sum_x (-1)^{\langle\alpha,x\rangle\oplus\langle\beta,f(x)\rangle} = \\ &= 2^{-n}\cdot \max_{\alpha,\beta\neq 0}\Big\{ |\,\{x \in \mathbb{F}_2^n : \langle \alpha, x\rangle \oplus \langle \beta, f(x)\rangle = 0\}\,| - |\,\{x \in \mathbb{F}_2^n : \langle \alpha, x\rangle \oplus \langle \beta, f(x)\rangle = 1\}\,|\Big\}. \end{aligned}$$

Since

$$|\,\{x \in \mathbb{F}_2^n : \langle \alpha, x\rangle \oplus \langle \beta, f(x)\rangle = 0\}\,| + |\,\{x \in \mathbb{F}_2^n : \langle \alpha, x\rangle \oplus \langle \beta, f(x)\rangle = 1\}\,| = 2^n,$$

these two definitions are strictly related, i.e. $corr(f) = 2\mathfrak{L}(f) - 1$.

[6] Since it is in general very hard to compute the probability of a differential, a common assumption in the literature is to approximate its probability with the sum of probabilities of some differential characterestics that define it.

- the maximum differential probability DP_{max} of the S-Boxes;
- the number of active S-Boxes.

Note that the maximum differential probability of an n-bit S-Box is at most 2^{-n+1}. Unfortunately, all known APN S-boxes have an odd number of input bits[7], with the exception of one 6-bit S-Box due to Dillon [BDMW10]. In more details, there is no (invertible) APN permutation of dimension 4 [LP07], while the question of finding an APN bijective (n, n)-function for even $n \geq 8$ is still open. Thus, the maximum differential probability of an n-bit S-Box for even $n \neq 6$ is "at most" 2^{-n+2}. This seems to suggest that one should take large S-Boxes.

Instead of spending most of its resources for looking for large S-Boxes with "good" properties, *the wide-trail strategy aims at designing the round transformation(s) in order to maximize the minimum number of active S-Boxes over multiple rounds.* Thus, in ciphers designed by the wide trail strategy, the idea is to look for linear layers that guarantee a large number of active S-Boxes over several rounds.

To achieve this goal, the concept of the *branch number* $\mathfrak{B}$ has been introduced in [Dae95] as a metric for the diffusion achieved by the round function:

> *The branch number $\mathfrak{B}(\lambda) \leq t+1$ of a linear transformation $\lambda : (\mathbb{F}_{2^n})^t \rightarrow (\mathbb{F}_{2^n})^t$ (and the resulting round function) is the minimum total number of active bundles in the input and output of the transformation, that is*
>
> $$\mathfrak{B}(\lambda) := \min_{x \neq 0} \{w_b(x) \oplus w_b(\lambda(x))\}$$
>
> *where $w_b(\cdot)$ denotes the number of non-zero bundles.*

Equivalently, the branch number $\mathfrak{B}(\lambda)$ corresponds to the minimum number of active S-Boxes over 2 rounds. Transformations λ which attain the maximum branch number

$$\mathfrak{B}(\lambda) = t+1$$

can be constructed using matrices of *Maximum Distance Separable* (MDS) Codes, as suggested by Vaudenay [Vau94]

> *A matrix $M \in \mathbb{F}_{2^n}^{t \times t}$ is called Maximum Distance Separable (MDS) matrix iff it has branch number $B(M)$ equal to $B(M) = t+1$.*

It is simple to observe that if a matrix M is MDS, then also its inverse M^{-1} is MDS.

For completeness, we remember the following theorem that characterize an MDS matrix.

Theorem 2 ([MS78]). *A $t \times t$ matrix M is an MDS matrix if and only if every square submatrix of M is nonsingular.*

It follows that all entries of an MDS matrix are non zero.

2-round AES. The previous result applies immediately to 2-round AES. First of all, AES S-Box has been chosen in order to have $DP_{max} = 2^{-6}$, that is the lowest possible[8]. Secondly, the 4×4 MixColumns matrix is an MDS matrix[9], which means that the *minimum* number of active S-Box over two consecutive rounds is equal to 5. As a result, the probability of each characteristic over 2 consecutive rounds of AES is at least

$$(DP_{max})^{\mathfrak{B}} = (2^{-6})^5 = 2^{-30}.$$

[7] For completeness, we mention that an almost bent function can only have an odd number of input bits. When n is even, almost bent functions do not exist, and the lowest possible linearity for an n-bit S-Box (where n is even) is not known. The best known value for n even is $\mathfrak{L} = 2^{\frac{-n+2}{2}}$, and this value is tight for a very few families of S-Boxes, including the inversion over $\mathbb{F}_2^n$ which is used e.g. in the AES.

[8] We mention that the linear correlation of the AES S-Box is 2^{-3}.

[9] Equivalently, this means that if x bytes are active in input, then *at least* $5 - x$ bytes are active in output.

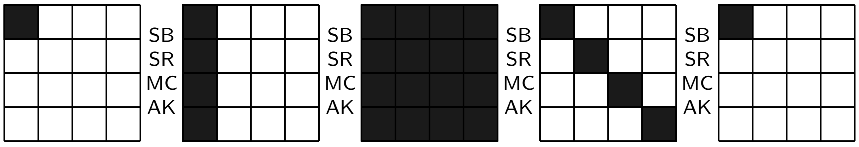


Figure 3.2.: 4-round AES characteristic with 25 active S-Boxes. A black byte denotes an active S-Box.

4-round AES. In order to apply the previous results on 4-round AES, the idea is to re-write it as a 2-round cipher. For this goal, we recall the *super-Sbox* notation [DR06] introduced by the designers

$$\textit{super-Sbox}(\cdot) = \text{S-Box} \circ ARK \circ MC \circ \text{S-Box}(\cdot), \tag{3.1}$$

which allows to rewrite[10] 4-round AES as

$$R^4(\cdot) = ARK \circ \textit{super-Sbox} \circ M' \circ \textit{super-Sbox} \circ ARK(\cdot)$$

where we omitted for simplicity the initial and the final ShiftRows operation (besides the final MixColumns operation) and where

$$M'(\cdot) = SR \circ ARK \circ MC \circ SR(\cdot).$$

Using Theorem 2 and since AES MixColumns matrix is MDS, it follows that M' is also an MDS matrix, which means that at least 5 *super-Sbox* are active over 4-round AES. Due to the previous result on 2-round AES, it follows that each active *super-Sbox* contains at least 5 active S-Boxes, for a total of 25 active S-Boxes over 4-round AES. As a result, the probability of each characteristic over 2 consecutive rounds of AES is at least

$$(2^{-6})^{25} = 2^{-150}.$$

Full AES. As we have seen, differential cryptanalysis exploits differentials rather than differential characteristics to set up key-recovery attacks. By definition, the probability of a differential is always greater (or equal) than the probability of a characteristic. The maximum probability of a 2-round AES differential has been computed (by exhaustive search) in [KMT01], and it is equal to $53/2^{34} \approx 2^{-28.27}$, while no result is known for 3 or more rounds.

On the other hand, since evaluating the true (maximum) differential probability (respectively, linear correlation) is computationally not practical for a typical block cipher, one natural solution is to try to upper bound these terms. This approach was chosen by Park *et al.* [PSC+02; PSLL03] who showed that the differential probability and linear correlation for 4-round AES are respectively bounded by 1.144×2^{-111} and 1.075×2^{-106}.

In conclusion, since each differential characteristic over 4-round AES has probability (much) lower than 2^{-128} and since AES is composed of (at least) 10 rounds, any differential attack is considered to be unfeasible [PSC+02; PSLL03]. A similar result holds for linear cryptanalysis.

3.3. Existing Cryptanalysis of AES

Besides differential and linear cryptanalysis, many other techniques have been proposed in the literature in order to set up distinguishers and key-recovery attacks. In the following, we recall the main ones.

[10] Note that $SR \circ \text{S-Box}(\cdot) = \text{S-Box} \circ SR(\cdot)$.

Table 3.1.: *Secret-key Distinguishers on round-reduced AES (which are independent of the secret-key).* All distinguishers in the table are independent of the details of the S-Box, of the details of key-schedule and of the MixColumns matrix (assuming branch number equal to 5). Data complexity is measured in minimum number of chosen plaintexts/ciphertexts CP/CC or/and adaptive chosen plaintexts/ciphertexts ACP/ACC which are needed to distinguish the AES permutation from a random permutation with probability (much) higher than 95%. Time complexity is measured in XOR operations (XOR), equivalent encryptions (E) or memory accesses (M) - using the common approximation 20 M $\approx$ 1-round E. Distinguishers proposed in our works are in bold.

Property	Rounds	Data	Computation	Reference
Truncated Differential	1 - 2	2 CP	1 XOR	[DR06]
Truncated Differential	3	$20 \simeq 2^{4.3}$ CP	$2^{7.6}$ M	*folklore*
Integral	3	2^{8} CP	2^{8} XOR	[DR98; DR02b]
Yoyo	4	2 CP + 2 ACC	1 XOR	[RBH17]
Boomerang	4	2 CP + 2 ACC	1 XOR	[Bir04]
Impossible Differential	4	$2^{16.25}$ CP	$2^{31.5}$ M $\approx 2^{25.18}$ E	[BK01]
Mixture Diff.	**4**	$\mathbf{2^{17}}$ **CP**	$\mathbf{2^{23.1}}$ **M** $\approx \mathbf{2^{16.75}}$ **E**	**[Gra18b]**
Integral	4	2^{32} CP	2^{32} XOR	[DR98; DR02b]
Yoyo	5	2^{12} CP + $2^{25.8}$ ACC	$2^{24.8}$ XOR	[RBH17]
Multiple-of-8	**5**	$\mathbf{2^{32}}$ **CP**	$\mathbf{2^{35.6}}$ **M** $\approx \mathbf{2^{29}}$ **E**	**[GRR17]**
Truncated Diff. (Variance)	**5**	$\mathbf{2^{34}}$ **CP**	$\mathbf{2^{37.6}}$ **M** $\approx \mathbf{2^{31}}$ **E**	**[GR18]**
Truncated Diff. (Mean)	**5**	$\mathbf{2^{48.96}}$**CP**	$\mathbf{2^{52.6}}$ **M** $\approx \mathbf{2^{46}}$ **E**	**[GR18]**
Prob. Mixture Diff.	**5**	$\mathbf{2^{52}}$ **CP**	$\mathbf{2^{71.5}}$ **M** $\approx \mathbf{2^{64.9}}$ **E**	**[Gra17b]**
Imp. Mixture Diff.	**5**	$\mathbf{2^{82}}$ **CP**	$\mathbf{2^{97.8}}$ **M** $\approx \mathbf{2^{91.1}}$ **E**	**[Gra17b]**
Threshold M.D.	**5**	$\mathbf{2^{89}}$ **CP**	$\mathbf{2^{98.1}}$ **M** $\approx \mathbf{2^{91.5}}$ **E**	**[Gra17b]**
Yoyo	6	$2^{122.83}$ ACC	$2^{121.83}$ XOR	[RBH17]

3.3.1. Integral Attack

In the paper presenting the block cipher Square [DKR97], a dedicated attack on reduced versions of Square is described. The attack is often referred to as the “Square” attack, but it is also called *Integral* [KW02] or Saturation attack [Luc01]. The attack exploits the byte-oriented structure of Square, and is also applicable to reduced versions of AES. This attack is a chosen plaintext attack, and it can be mounted independently of the choice of the S-Box, of the key-schedule and of the details of the linear operations. In an integral attack, one considers a set Λ of input texts with the following characteristics:

- it sums to 0 and its values are fixed in some bit/byte/word positions and take all possible combinations of values in the other bit/byte/word positions;
- the sum of the corresponding encrypted values is equal to 0 (at least in some bits):

$$\bigoplus_{t\in\Lambda} t = \bigoplus_{t\in\Lambda} R^r(t) = 0.$$

Since for a random permutation the same event happens with probability lower than 1 (e.g. with prob. 2^{-128} for a random permutation on 128 bits), this zero-sum property can be exploited in order to distinguish a cipher from a random permutation.

Table 3.2.: *Comparison of attacks on round-reduced AES-128.* Data complexity is measured in number of required chosen plaintexts/ciphertexts (CP/CC) or/and adaptive chosen plaintexts/ciphertexts ACP/ACC . Time complexity is measured in round-reduced AES encryption equivalents (E) - the number in the brackets denotes the precomputation cost (if not negligible). Memory complexity is measured in texts (16 bytes). Attacks proposed in our works are in bold.

Attack	Rounds	Data	Computation	Memory	Reference
MitM	5	8 CP	2^{64}	2^{56}	[Der13, Sec. 7.5.1]
Imp. Polytopic	5	15 CP	2^{70}	2^{41}	[Tie16a]
Partial Sum	5	2^{8} CP	2^{38}	small	[Tun12]
Integral (EE)	5	2^{11} CP	$2^{45.7}$	small	[DR02b]
Yoyo	5	$2^{11.3}$ ACC	2^{31}	small	[RBH17]
Mixture Diff.	5	$2^{22.5}$ CP	$2^{22.5}$	2^{20}	[BDK+18]
Imp. Differential	5	$2^{31.5}$ CP	2^{33} $(+2^{38})$	2^{38}	[BK01]
Integral (EB)	5	2^{33} CP	$2^{37.7}$	2^{32}	[DR02b]
Variance	**5**	**2^{33} CP**	**$2^{64.2}$**	**2^{32}**	**[GR18]**
Mixture Diff.	**5**	**$2^{33.6}$ CP**	**$2^{33.3}$**	**2^{34}**	**[Gra18b]**
Multiple-of-n	**5**	**$2^{33.6}$ CP**	**2^{48}**	**2^{32}**	**[GRR17]**
Trunc. Diff.	**5**	**2^{35} CP**	**$2^{69.2}$**	**2^{32}**	**[GR18]**
Boomerang Attack	5	2^{39} CP/ACC	2^{39}	2^{33}	[Bir04]
MitM	6	13 CP	2^{120}	2^{96}	[Der13, Sec. 7.5.2]
MitM	6	2^{8} CP	$2^{106.2}$	$2^{106.2}$	[Der13, Sec. 7.3.3]
Mixture Diff.	6	$2^{27.5}$ CP	2^{81}	$2^{27.5}$	[BDK+18]
Partial Sum	6	2^{32} CP	2^{42}	2^{40}	[Tun12]
Integral	6	2^{35} CP	$2^{69.7}$	2^{32}	[DR02b]
Boomerang Attack	6	2^{71} CP/ACC	2^{71}	2^{33}	[Bir04]
Prob. Diff. Struc.	**6**	**$2^{72.8}$ CP**	**2^{105}**	**2^{33}**	**[Gra17b]**
Imp. Differential	6	$2^{91.5}$	2^{122}	2^{89}	[CKK+02]
MitM	7	2^{32} CP	$2^{126.5}$	$2^{126.5}$	[Der13, App. 7.B.6]
MitM	7	2^{97} CP	2^{99}	2^{98}	[DF13]
Imp. Differential	7	$2^{106.2}$ CP	$2^{110.2}$	$2^{90.2}$	[MDRM10]
Herds Attack	7	$(2^{128}-2^{119})$ CP	2^{120}	(?)	[FKL+00]
Biclique	8	2^{88} CP	$2^{125.34}$	2^{8}	[BKR11]
Biclique	8	2^{127} CP	$2^{125.64}$	2^{32}	[BKR11]
Biclique	10	2^{88} CP	$2^{126.18}$	2^{8}	[BKR11]

MitM: Meet-in-the-Middle, EE: Extension at End, EB: Extension at Beginning

To describe such attack/distinguisher on round-reduced AES in more details, we first recall some notations largely used in the literature. Given a set of texts $\Lambda = \{t^i\}_i$, we say that the bytes in position (j, k) for $0 \leq j, k \leq 3$ are

constant (C): $t_{j,k} = s_{j,k}$ for each $t, s \in \Lambda$;

active (A): $t_{j,k} \neq s_{j,k}$ for each $t, s \in \Lambda$;

balance (B): $\bigoplus_{t \in \Lambda} t_{j,k} = 0$;

unknown (?): if no one of the previous property is satisfied.

Note that if a byte is constant or active, then by definition it is also balance.

3-round (secret-key) Distinguisher. Consider a Λ-set of 2^8 chosen plaintexts in which only one byte is active. It is possible to prove that all bytes of the corresponding ciphertexts after 3-round AES are balanced independently of the secret key:

$$\begin{bmatrix} A & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \end{bmatrix} \xrightarrow{R(\cdot)} \begin{bmatrix} A & C & C & C \\ A & C & C & C \\ A & C & C & C \\ A & C & C & C \end{bmatrix} \xrightarrow{R(\cdot)} \begin{bmatrix} A & A & A & A \\ A & A & A & A \\ A & A & A & A \\ A & A & A & A \end{bmatrix} \xrightarrow{R(\cdot)} \begin{bmatrix} B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \end{bmatrix}.$$

4-round (secret-key) Distinguisher. Consider a Λ-set of 2^{32} chosen plaintexts in which one diagonal[11] is active. It is possible to prove that all bytes of the corresponding ciphertexts after 4-round AES are balanced independently of the secret key.

To show this fact, note that such set Λ is mapped into a Λ'-set of 2^{32} chosen plaintexts in which one column is active. Since the set Λ' can be seen as the union of 2^{24} sets of 2^8 texts in which only one byte (e.g. the first one) is active, the result follows immediately due to the previous integral distinguisher on 3-round AES. Again, since for a random permutation the same event happens with prob. 2^{-128}, it is possible to distinguish 4-round AES from it.

Key-Recovery Attacks. The previous distinguishers can be extended at the beginning and/or at the end into key-recovery attacks for up to 6-round AES-128 and for up to 7-round AES-192/256. The computational complexities of these attacks has then been improved by Ferguson *et al.* in [FKL+00], using the *partial sum* technique. Besides that, a further extension for up to 7-round AES-128 and for up to 8-round AES-192/256 - called the Herds attack - have been proposed by N. Ferguson *et al.* in [FKL+00]. On the other hands, such attacks require $2^{128} - 2^{119}$ chosen plaintexts, that is almost the full codebook.

3.3.2. Truncated Differential Attack

The concept of truncated differential was initially proposed by L. Knudsen in [Knu94].

As we have just seen, differential attacks exploit the fact that pairs of plaintexts with certain differences yield other differences in the corresponding ciphertexts with a non-uniform probability distribution. In some cases, such attack can be improved (both from the complexity point of view or/and by the number of rounds that can be attacked) by considering differences that are not fully specified, but only specified for selected bits/bytes, while the exact difference in the remaining bits/bytes is ignored. Truncated differential attack/distinguisher is a variant of classical differential attack/distinguisher in which the attacker can predict *only part of the difference* between pairs of texts.

About AES, it is possible to set up secret-key distinguishers - which are independent of the key and of the details of the S-Box - for up to 3-round AES. In particular, assume that the last MixColumns is omitted:

- given two plaintexts that differs in the i-th diagonal, then the corresponding ciphertexts after 2-round AES are equal in all bytes expect for the ones that lie in the i-th anti-diagonal with prob. 1. Since for a random permutation the same event happens with prob. 2^{-96}, it is possible to distinguish the two cases;

[11] The i-th diagonal of a 4×4 matrix A is defined as the elements that lie on row r and column c such that $r - c = i \mod 4$. The i-th anti-diagonal of a 4×4 matrix A is defined as the elements that lie on row r and column c such that $r + c = i \mod 4$.

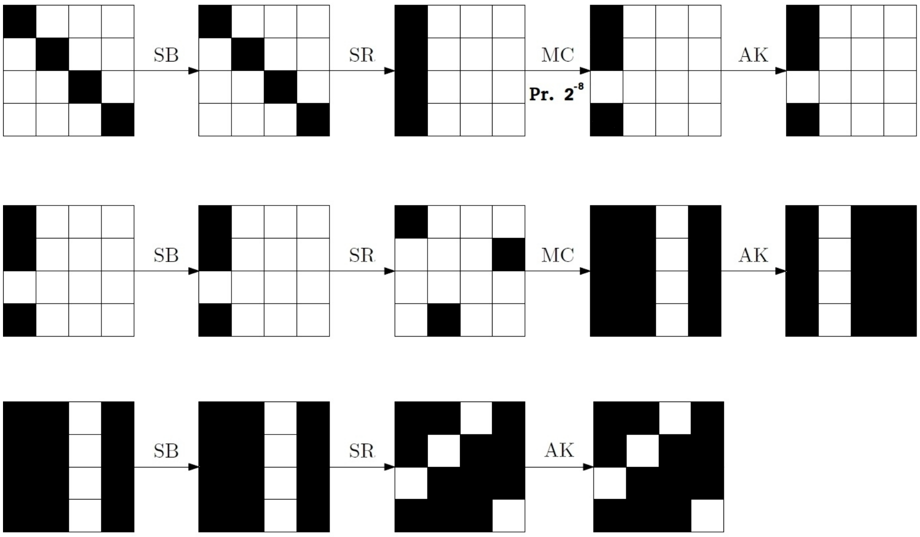


Figure 3.3.: Truncated Differential on 3-round AES. A black byte denotes a byte with non-zero difference, while a white byte denotes a byte with zero-difference.

- given two plaintexts that differs in the i-th diagonal, then the corresponding ciphertexts after 3-round AES are equal in the j-th anti-diagonal (for a certain $j \in \{0, 1, 2, 3\}$ fixed) with prob. 2^{-8}. Since for a random permutation the same event happens with prob. 2^{-32}, it is possible to distinguish the two cases.

More details are given in the following using the subspace trail terminology.

Such distinguishers can be exploited to set up key-recovery attacks on round-reduced AES.

3.3.3. Impossible Differential Attack

In standard differential cryptanalysis, a common approach is to use differentials that have a sufficiently high probability to distinguish the cipher from a random permutation. On the other hand, another possibility is to use differentials that have zero probability. Such differentials are called *impossible differentials*. Impossible differentials were independently developed by Biham, Biryukov, and Shamir [BBS99] and by Knudsen [Knu98].

One of the most important class of impossible differentials is impossible truncated differentials, i.e. truncated differentials of probability zero. Such impossible truncated differentials can often be constructed by combining two probability-one truncated differentials that do not match in the middle.

For the AES case, it is possible to set up secret-key distinguishers based on impossible truncated differentials which are independent of the secret key and of the details of the S-Box for up to 4 rounds [BK01]. E.g., given a pair of texts that differ in only one diagonal, the corresponding ciphertexts after 4-round can not be equal in any of the four anti-diagonal (assuming the last MixColumns is omitted). Since the same event happens with non-zero probability for a random permutation, it is possible to distinguish the two cases. More details are given in the following using the subspace trail terminology.

Key-Recovery Attacks. Together with the impossible differential distinguisher on 4-round AES, Biham and Keller [BK01] proposed the first impossible differential attack on 5-round AES. This attack has been improved in [CKK+02], where authors also presented the first attack on 6 rounds. In [Pha04], the first impossible differential attack on 7-round AES-192 and AES-256 is presented. Later, based on different impossible differentials, new 7-round impossible differential attacks – also on

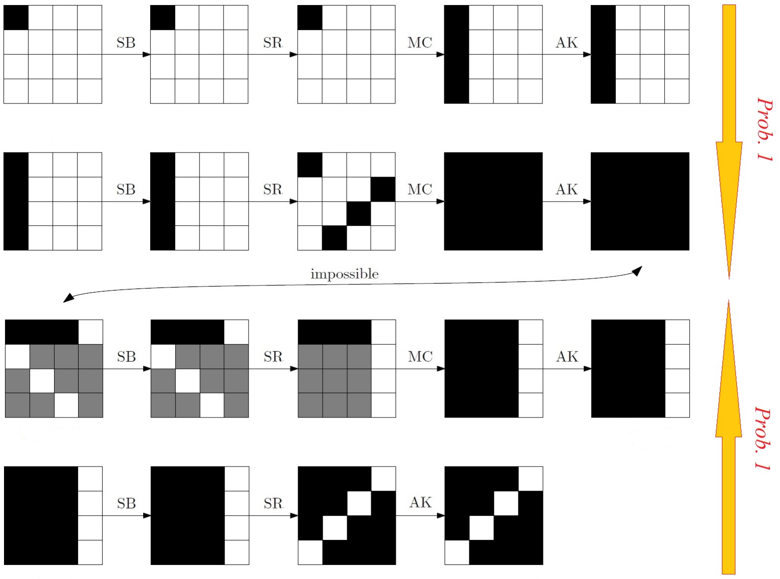


Figure 3.4.: Impossible Differential on 4-round AES. A black byte denotes a byte with non-zero difference, while a white byte denotes a byte with zero-difference. A gray byte denotes a byte with an unknown difference.

AES-128 – were presented in [BA08]. Using various techniques, including the early abort approach and key-schedule considerations, the attack has then been improved in [LDKK08], resulting in the best impossible differential attacks on AES-192 and AES-256, and in [MDRM10], resulting in the best impossible differential attacks on AES-128 so far. Finally, only recently Boura *et al.* [BLNS18] improved it into its best currently known variant which breaks 7-round AES with an overall complexity of about 2^{107}

3.3.4. Meet-in-the-Middle Attacks

The Meet-in-the-Middle (MitM) attack is a generic attack applicable to a large variety of cryptographic primitives. The main idea is to split the cipher into two independent parts and use a time-memory trade-off for a more efficient attack. In more details, assume to split the cipher E into two parts $E(\cdot) = E_2 \circ E_1(\cdot)$. Roughly speaking, given a plaintext-ciphertext pair (p, c) obtained under the secret key K, the attacker partially guesses the secret key and check if

$$p \xrightarrow{E_1(\cdot)} \overrightarrow{v} \overset{?}{=} \overleftarrow{v} \xleftarrow{E_2(\cdot)} c.$$

If there is no match in the middle, it turns out that the guessed key is wrong.

While the original Meet-in-the-Middle attack was very generic, many subsequently improvements to it have been suggested by using the underlying structure of the cryptographic primitive. Since a basic Meet-in-the-Middle attack requires only the information-theoretical minimum of plaintext-ciphertext pairs, it can (potentially) be the most practical in terms of data complexity.

Meet-in-the-Middle Attacks on AES. The original Meet-in-the-Middle attack against AES was proposed by Demirci and Selçuk at FSE 2008 [DS08], and it relies on particular sets called Λ-sets equal to the ones introduced by Daemen *et al.* against the block cipher Square. More specifically, they show that on 4 rounds, the value of each byte of the ciphertext can be described by a function of the active byte parameterized by 25 8-bit parameters (reduced to 24 in [DTÇB09]). Several improvements for the attack were then proposed at Asiacrypt 2010 by Dunkelman *et al.* [DKS10], and they mainly include *(1st)* the differential enumeration technique and *(2nd)* a clever and powerful memory/data trade-off that does not change the time. Then at Eurocrypt 2013, Derbez *et al.* [DFJ13] showed that this technique leads to much better attacks than expected by Dunkelman *et al.*, and reached the best known attacks against 7-round AES-128 and 9-round AES-256 in the single-key model. Next, at FSE 2013, Derbez and Fouque [DF13] generalized the attack of Demirci and Selçuk by searching a match on some equation and not only on the byte state.

Bycicle Attack. Meet-in-the-Middle attacks on block ciphers have also great potential if enhanced with bicliques. The biclique concept was first introduced for hash cryptanalysis by Savelieva *et al.* [KRS12]. Its approach led to the best preimage attacks on the SHA family of hash functions so far, including the attack on 50 rounds of SHA-512, and the first attack on a round-reduced Skein hash function [KRS12].

A biclique is characterized by its length (number of rounds covered) and dimension. The dimension is related to the cardinality of the biclique elements and is one of the factors that determines the advantage over brute force. The total cost of the key search with bicliques depends on two main contributors, namely the cost of constructing the bicliques and the cost of the matching computations.

Biclique cryptanalysis [BKR11] successfully applies to all full versions of AES and compared to brute force provides a computational advantage of about a factor 3 to 5, depending on the version.

3.3.5. Interpolation and Algebraic Attacks

Algebraic attacks model a cryptographic primitive (such as a block cipher) as a system of equations. By applying (algebraic) transformations to these equations, these attacks (attempt to) recover information about the secret of the primitive (the key).

One example of algebraic attack is the interpolation attack, introduced by T. Jakobsen and L. Knudsen in [JK97]. In this attack, the attacker constructs a polynomial corresponding to the encryption function without knowledge of the secret key. If an adversary can construct such a polynomial then for any given plaintext the corresponding cipher-text can be produced without knowledge of the secret key.

In more details, let e.g. $E_k(\cdot) : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ be an encryption function. For a randomly fixed key k, the polynomial $P(x)$ representing $E_k(x)$ can be constructed using e.g. Lagrange's theorem, where x is the indeterminate corresponding to the plaintext. If the polynomial has degree d then we can find it using Lagrange's formula

$$P(x) = \sum_{i=0}^{d} y_i \prod_{j \neq i} \frac{x - x_i}{x_j - x_i} \tag{3.2}$$

where $E_k(x_i) = y_i$ for $0 \leq i \leq d$. This method can be extended to a key recover attack. The attack proceeds by simply guessing the key of the final round, decrypting the ciphertexts and constructing the polynomial for $r-1$ rounds. An extra pair of texts (p, c) is then used to check the guessed key (note that the equality $P(p) = R_k^{-1}(c)$ is satisfied by the secret key).

About AES, the complicated expression of the S-Box in $GF(2^8)$

$$\text{S-Box}(x) = 63 \oplus 8Fx^{127} \oplus B5x^{191} \oplus 01x^{223} \oplus F4x^{239} \oplus 25x^{247} \oplus F9x^{251} \oplus 09x^{253} \oplus 05x^{254}$$

together with the effect of the mixing and transposition steps, prohibis interpolation attack on more than a few rounds[12].

For completeness, we mention that a detailed algebraic analysis of AES has been performed in [CP02], where authors proposed an attack that can potentially break full AES-128 faster than brute force. Such attack works by trying to express the entire algorithm as multivariate quadratic

[12] As a concrete example, the algebraic representation for 10-round AES found by N. Ferguson *et al.* [FSW01] would count 2^{50} terms.

polynomials, and it used the so called “XLS technique” to solve it. On the other hand, a complete analysis of the XSL algorithm presented at Asiacrypt 2005 [CL05] led to the result that - in its current form - the XSL algorithm does not provide an efficient method for solving the AES system of equations.

3.3.6. Higher-Order Differential

Higher-order differentials consider the difference of more than two texts. The idea – first introduced by Lai [Lai94] without a concrete application however – was used by Knudsen [Knu94] in 1995 to describe higher-order differentials that can be used to break ciphers (with low algebraic degree) which are secure against standard differential cryptanalysis.

For simplicity, we limit ourselves here to present the idea of higher-order differential cryptanalysis for bit-based ciphers only.

Algebraic Normal Form and Algebraic Degree. Each function $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ admits a unique representation as a polynomial of degree smaller than 2^n:

$$F(x) = \bigoplus_{i=0}^{2^n-1} \varphi_i \cdot x^i, \qquad \varphi_i \in \mathbb{F}_{2^n} \tag{3.3}$$

where x is a variable in $\mathbb{F}_{2^n}$. Such a function is *linear* if and only if it can be expressed as

$$F^L(x) = \bigoplus_{i=0}^{n-1} \varphi_i \cdot x^{2^i}, \qquad \varphi_i \in \mathbb{F}_{2^n},$$

while it is affine if it is the sum of a linear function and a constant.

At the same time, the function $F(\cdot)$ admits a unique representation – called “algebraic normal form” – as a polynomial in n variables

$$F(x) = \bigoplus_{u \in \mathbb{F}_2^n} \varphi(u) \cdot \left(\prod_{i=1}^{n} x_i^{u_i} \right)$$

where $u = (u_n, u_{n-1}, ..., u_1) \in \mathbb{F}_2^n$.

Definition 1. *Let $f(\cdot)$ a boolean function of n variables. The Algebraic Normal Form (ANF) of $f(\cdot)$ is defined as*

$$f(x) = \bigoplus_{u \in \mathbb{F}_2^n} \mu_f(u)\mathbf{x}^{\mathbf{u}}$$

where $\mathbf{x}^{\mathbf{u}} = x_0^{u_0} \cdot x_1^{u_1} \cdot ... \cdot x_{n-1}^{u_{n-1}}$, $\mu_f(u) = \bigoplus_{x \preceq u} f(x)$ and $x \preceq u$ iff $x_i \leq u_i$ for each $0 \leq i < n$.

The (algrebraic) degree of the ANF of a function $f(\cdot)$ is defined as the maximum of the degrees of the monomials of its ANF

$$\deg(f) = \max\{hw(\mathbf{u}) \,|\, \mu(u) = 1\}$$

where $hw(\cdot)$ is the hamming weight[13].

[13] Given $n \in \mathbb{N}$, its hamming weight is defined as $hw(n) = \sum_{i=0}^{\lfloor \log_2(n) \rfloor} n_i$ where $n = \sum_{i=0}^{\lfloor \log_2(n) \rfloor} n_i \cdot 2^i$ and $n_i \in \{0, 1\}$.

As showed e.g. in [CCZ98], these two representations are equivalent. Let α be a *primitive element*[14] in $\mathbb{F}_{2^n}$, it follows that each $x \in \mathbb{F}_{2^n}$ can be rewritten as $\bigoplus_{j=1}^{n} x_j \cdot \alpha^{j-1}$. Thus

$$F(x) = F\left(\bigoplus_{j=1}^{n} x_j \cdot \alpha^{j-1}\right) = \bigoplus_{i=0}^{2^n-1} \varphi_i \left(\bigoplus_{j=1}^{n} x_j \cdot \alpha^{j-1}\right)^i =$$

$$= \bigoplus_{i=0}^{2^n-1} \varphi_i \left(\bigoplus_{j=1}^{n} x_j \cdot \alpha^{j-1}\right)^{\sum_{s=0}^{n-1} i_s \cdot 2^s} = \bigoplus_{u \in \mathbb{F}_2^n} \varphi(u) \cdot \left(\prod_{i=1}^{n} x_i^{u_i}\right).$$

Note that $(\bigoplus_i x_i)^{2^s} = \bigoplus_i x_i^{2^s}$ (since $f(x) = x^{2^s}$ is a linear function).

As showed in [CCZ98], the algebraic degree of $F(x) = \bigoplus_{i=0}^{2^n-1} \varphi_i \cdot x^i$ is equal to the maximum of the hamming weight of its exponents.

Definition 2 ([CCZ98])**.** *Let $F(\cdot)$ be a polynomial given by the expression (3.3). $F(\cdot)$ has algebraic degree $\delta(F)$ if δ is the maximum hamming weight of its exponents:*

$$\delta = \max_{0 \le i \le 2^n-1} \{hw(i) \,|\, \varphi_i \neq 0\}$$

In particular, if F is a polynomial of degree d, it follows that $\delta = \max_{0 \le i \le d < 2^n-1} \{hw(i) \,|\, \varphi_i \neq 0\}$.

Higher-Order Differential Attack. Following [Lai94], the derivative Δ_α of a boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ in the direction of $\alpha \in \mathbb{F}_n^2$ is defined as

$$\Delta_\alpha f(x) := f(x \oplus \alpha) \oplus f(x)$$

where x is the input bit vector. This derivative shares many properties with the standard derivative over the real numbers: it is linear, it satisfies (a variant of) the product rule and importantly it reduces the degree of the function by at least 1.

In a similar way, it is possible to define the d-th order derivative of the (vectorial) Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ (by $\alpha_1, \ldots, \alpha_d \in \mathbb{F}_2^n$) as

$$\Delta^{(d)}_{\alpha_1,\ldots,\alpha_d} f(x) := \Delta_{\alpha_d} \cdots \Delta_{\alpha_1} f(x).$$

Due to the product rule and due to the definition of the boolean derivative, two important properties can be highlighted:

- first, the algebraic degree of $\Delta^{(d)} f$ is
$$\deg \Delta^{(d)}_{\alpha_1,\ldots,\alpha_d} f(x) \le \deg f - d\,;$$
- secondly, the following equivalence
$$\Delta_{\alpha_d} \cdots \Delta_{\alpha_1} f(x) = \bigoplus_{\alpha \in A} f(x \oplus \alpha)\,,$$
holds, where the sum ranges over all 2^d elements of the span
$$A = \langle \alpha_1, \ldots, \alpha_d \rangle = \{\lambda_1 \alpha_1 \oplus \ldots \oplus \lambda_d \alpha_d \mid \lambda \in \mathbb{F}_2^d\}\,.$$

Both these two properties are exploited by higher-order differential. In more details, higher-order differential cryptanalysis exploits the fact that given a subspace A whose dimension d satisfies $d \ge \deg f + 1$, then for any offset α:

$$\bigoplus_{x \in A \oplus \alpha} f(x) = 0.$$

[14] An element $\alpha \in \mathbb{F}_{2^n}$ is "primitive" if for each $\beta \in \mathbb{F}_{2^n}$ there exists j s.t. $\alpha^j = \beta$.

How to Prevent Higher-Order Differential? To prevent such attacks, ideally one would like to be able to make a statement such as "After r rounds there is no output bit and no input subspace of dimension d' s.t. the derivative of the polynomial representation of the output bit with respect to this subspace is the zero-polynomial." To achieve such goal, one needs to estimate the *growth of the degree.*

Consider a SPN cipher defined over $(\mathbb{F}_{2^n})^t$ where $N = t \cdot n$. First of all, note that the growth of the degree is independent of the linear layer. Secondly, observe that the algebraic degree does not increase if several non-linear functions (namely, S-Boxes) are applied in parallel. Thus, denoting by d the degree of the S-Box in its algebraic representation in $GF(2^n)$, it follows that the algebraic degree of the cipher after r rounds is bounded from above by d^r. It is furthermore generally bounded from above by $N - 1$ since the cipher is a permutation.

A better and certainly more realistic upper bound was found by Boura, Canteaut, and De Canniére:

Proposition 1 ([BCC11]). *Let F be a function from $\mathbb{F}_2^N$ into $\mathbb{F}_2^N$ corresponding to the concatenation of t smaller balanced*[15] *S-Boxes $S_1, ..., S_t$ defined over $\mathbb{F}_2^n$ Then, for any function G from $\mathbb{F}_2^N$ into $\mathbb{F}_2^N$, we have*

$$\deg(G \circ F) \leq \min\left\{\deg(G) \cdot \deg(F), N - \frac{N - \deg(G)}{n-1}\right\}. \tag{3.4}$$

Moreover, if $n \geq 3$ and all S-Boxes have degree at most $n - 2$, we have

$$\deg(G \circ F) \leq \min\left\{\deg(G) \cdot \deg(F), N - \frac{N - \deg(G)}{n-2}\right\}. \tag{3.5}$$

These bounds can be exploited in order to compute the algebraic degree of the encryption/decryption function after r rounds, and so to compute the minimum number of rounds necessary to guarantee security against higher-order differential attacks.

About AES, full AES can be considered secure against higher-order differential attacks due to the same considerations made for the interpolation attack.

Division Property. A generalization of integral and higher-order differential distinguisher – called "division property" – has been recently introduced by Todo at Eurocrypt 2015 [Tod15b].

Let $u = (u_1, ..., u_n), x = (x_1, ..., x_n)$ be vectors in $\mathbb{F}_2^n$ and let u^x be defined as $\prod_{i=1}^n u_i^{x_i}$. A set $X \subseteq \mathbb{F}_2^n$ has the division property $\mathcal{D}_k^n$ for $1 \leq k \leq n$ if

$$\forall u \in \mathbb{F}_2^n \text{ s.t. } wt(u) < k : \qquad \bigoplus_{x \in X} x^u = 0$$

where $wt(\cdot)$ denotes the Hamming weight.

Using the notation proposed for integral attack, it is possible to show that a set X has the division property $\mathcal{D}_2^n$ if and only if the set X is balanced. Moreover, if a set X is active, then it has the division property $\mathcal{D}_n^n$ (vice-versa is in general not true). The novelty here is that it introduces intermediate properties $\mathcal{D}_k^n$ for $3 \leq k \leq n - 1$ which do not appear in classical integral attacks. This allows to study the propagation of $\mathcal{D}_n^k$ over multiple rounds, capturing information resulting from the algebraic degree of the round function. In such a sense, division property can be seen as a generalization of higher-order differential.

Since it is not possible to improve the results just proposed by integral and higher-order differential attacks on AES using the division property and since we do not consider the division property in the following of this Thesis, we refer to [Tod15b; BC16] for more details about this topic.

[15] Any function $f(\cdot) : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is *balanced* if each element in $\mathbb{F}_2^m$ has exactly 2^{n-m} preimages under $f(\cdot)$.

3.3.7. Link among different Cryptanalytic Tools

Along with the growing of the list of cryptanalytic tools, the question whether there are direct links or any connections among different tools has drawn much attention of the cryptographic research community, since such relations can be used to compare the effectiveness of different tools as well as to improve cryptanalytic results on block ciphers.

The first theoretical link between differential and linear cryptanalysis was presented by Chabaud and Vaudenay in [CV94]. After that, many attempts have been made to establish further relations among various cryptanalytic tools. In [SLQL10], Sun *et al.* proved that from an algebraic view, integral cryptanalysis can be seen as a special case of the interpolation attack. In [Lea11], Leander stated that statistical saturation distinguishers are averagely equivalent to multidimensional linear distinguishers. In [BLNW12], Bogdanov *et al.* showed that an integral implies a zero correlation linear hull unconditionally, a zero correlation linear hull indicates an integral distinguisher under certain conditions, and a zero correlation linear hull is actually a special case of multidimensional linear distinguishers.

Later on, in [BN13; BLN14], Blondeau and Nyberg further analyzed the link between differential and linear cryptanalysis and demonstrated some new insights on this link to make it more applicable in practice. They established new formulas between the probability of truncated differentials and the correlation of linear hulls. Moreover, they claimed that the existence of a zero correlation linear hull is equivalent to the existence of an impossible differential in some specific cases. This link has been proved in [SLR+15], where Sun *et al.* established the link between impossible differential cryptanalysis and integral cryptanalysis. Moreover, in there they showed that constructing impossible differentials of a structure is equivalent to constructing zero correlation linear hulls of the dual structure.

Finally, in [BN14], Blondeau and Nyberg proposed the link between truncated differential and multidimensional linear approximation, and then applied this link to explore the relations between the complexities of chosen-plaintext and known-plaintext distinguishing/key recovery attacks of differential and linear types. Moreover, they showed that statistical saturation cryptanalysis is indeed equivalent to truncated differential cryptanalysis, which could be used to estimate the data requirement of the statistical saturation key recovery attack.

3.3.8. Boomerang and Yoyo Attacks

W.r.t. previous attacks/distinguishers, boomerang and yoyo attacks require adaptive chosen plaintexts/ciphertexts besides known/chosen plaintexts/ciphertexts.

Boomerang Attack

Boomerang attacks [Wag99] allow to analyze a given cryptographic transform that lacks long differentials with sufficient probability, but for which short differentials with high probabilities exist. Say, $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ is a cipher that can be decomposed into parts $E(\cdot) = E_2 \circ E_1(\cdot)$ such that there exist a differential $\alpha \to \beta$ with probability $prob_1$ over E_1 and a differential $\gamma \to \delta$ with probability $prob_2$ over E_2. In the following, let $D(\cdot) \equiv E^{-1}(\cdot) \equiv D_1 \circ D_2(\cdot)$. A boomerang distinguisher follows the procedure:

1. Choose a plaintext pair (p, p'), with $p' = p \oplus \alpha$, and ask for its corresponding ciphertext (c, c') through E;

2. Compute $d = c \oplus \delta$ and $d' = c' \oplus \delta$ to obtain the ciphertext pair (d, d'), and ask for its corresponding plaintext (q, q'). This is also called a δ-shift;

3. Check if $q \oplus q' = \alpha$.

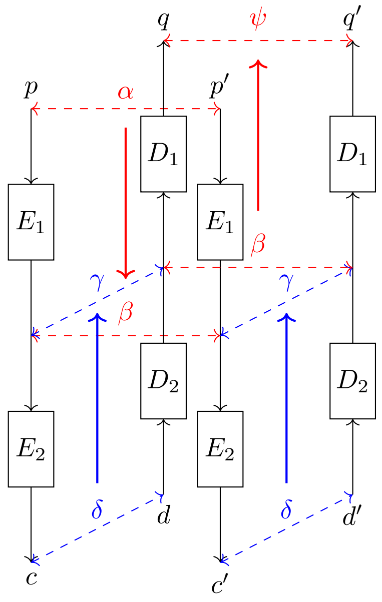


Figure 3.5.: Schematic illustration of a boomerang construction

The probability that $q \oplus q' = \alpha$ for a block cipher is (roughly) approximated by $(prob_1 \cdot prob_2)^2$ since the trails must hold for both pairs. If the probability that $q \oplus q' = \alpha$ is different than from the corresponding probability a random permutation - which is given by 2^{-n}, then it is possible to distinguish distinguisher or key-recovery attack. Figure 3.5 illustrates the boomerang construction schematically.

Such probability can be increased by considering all possible internal trails $\alpha \to \beta'$ and $\gamma \to \delta'$, where β' and δ' are not fixed. This allows to increase the probability that $q \oplus q' = \alpha$ for a block cipher, which is now equal to

$$\sum_{\beta'} \sum_{\gamma'} Prob^2(\alpha \to \beta') \cdot Prob^2(\gamma \to \delta').$$

Boomerang Attacks on round-reduced AES. A boomerang distinguisher on 4-round AES - independent of the secret-key - was first presented in [Bir04], by combining two 2-round truncated differential with prob. 1. Such distinguisher requires approximately $2^{17.3}$ chosen plaintexts, $2^{17.3}$ adaptive chosen ciphertexts and a computational cost of $2^{19.65}$ table look-ups. Key-recovery attacks on 5- and 6-round AES that exploit such distinguisher have been proposed in the same paper.

Yoyo Attack on AES

Yoyo game cryptanalysis was introduced by Biham *et al.* in [BBD+98] for cryptanalysis of 16 rounds of SKIPJACK. Yoyo games are similar to Boomerang attacks, and they are based on adaptively making new pairs of plaintexts and ciphertexts that preserve a certain property inherited from the original pair.

At Asiacrypt 2016, the first key-independent yoyo-distinguishers for 4- and 5-rounds of AES has been proposed by Rønjom *et al.* [RBH17]. These distinguishers beat previous records and require respectively 2 and $2^{25.8}$ adaptive chosen ciphertexts, and essentially zero computation except for observing differences. In addition, authors present the first key-independent distinguisher for 6-rounds AES based on yoyos that preserve impossible zero differences in plaintexts and ciphertexts. This

Table 3.3.: *Comparison of low-data attacks on round-reduced AES-128.* Data complexity is measured in number of required known/chosen plaintexts (KP/CP). Time complexity is measured in round-reduced AES encryption equivalents (E), while memory complexity is measured in plaintexts (16 bytes). The case in which the MixColumns operation is omitted in the last round is denoted by "r.5 rounds", that is r full rounds and the final round. Attacks proposed in our works are in bold.

Attack	Rounds	Data	Computational	Memory	Reference
G&D-MitM	2.5	2 KP	2^{80}	2^{80}	[BDF11]
TrD	**2.5 - 3**	**2 CP**	$\mathbf{2^{31.6}}$	$\mathbf{2^{8}}$	**[GRR16]**
G&D-MitM	2.5	2 CP	2^{24}	2^{16}	[BDF11]
G&D-MitM	3	2 CP	2^{16}	2^{8}	[BDF11]
TrD	**2.5 - 3**	**3 CP**	$\mathbf{2^{11.2}}$	–	**[GRR16]**
G&D-MitM	3	3 CP	2^{8}	2^{8}	[BDF11]
TrD	**2.5 - 3**	**3 CP**	$\mathbf{2^{5.7}}$	$\mathbf{2^{12}}$	**[GRR16]**
TrD (EE)	**3.5 - 4**	**2 CP**	$\mathbf{2^{96}}$	–	**[GRR16]**
G&D-MitM	4	2 CP	2^{88}	2^{8}	[BDF11]
G&D-MitM	4	2 CP	2^{80}	2^{80}	[BDF11]
G&D-MitM	3.5	2 CP	2^{72}	2^{72}	[BDF11]
TrD (EE)	**3.5 - 4**	**3 CP**	$\mathbf{2^{74.7}}$	–	**[GRR16]**
G&D-MitM	4	3 CP	2^{72}	2^{8}	[BDF11]
TrD (EE)	**3.5 - 4**	**3 CP**	$\mathbf{2^{69.7}}$	$\mathbf{2^{12}}$	**[GRR16]**
G&D-MitM	4	4 CP	2^{32}	2^{24}	[BDF11]
ImpPol	3.5 - 4	8 CP	2^{38}	2^{15}	[Tie16a]
TrD (EB)	**3.5 - 4**	**24 CP**	$\mathbf{2^{35.1}}$	$\mathbf{2^{17}}$	**[GRR16]**

G&D: Guess & Det., D: Diff., MitM: Meet-in-the-Middle, TrD: Truncated Differential, ImpPol: Imp. polytopic, EE: Extension at End, EB: Extension at Beginning.

distinguisher requires an impractical amount of $2^{122.83}$ adaptive chosen plaintext/ciphertext pairs and essentially no computation apart from observing the corresponding differences.

Due to the similarity with "Mixture Differential" cryptanalysis, more details are given in the following using the subspace trail terminology.

3.3.9. "Low-Data" and Polytopic Attacks

As already recalled in the introduction, a common approach of the cryptanalysis community is to consider attacks on reduced-round variants of block ciphers. Here, the usual goal of the adversary is to maximize the number of rounds that can be broken, using less data than the entire codebook and less time than exhaustive key search. Attacks following such an approach are of importance, since they ensure that the block ciphers are strong enough and because they help to establish the security margins offered by the cipher. However, aiming for the highest number of rounds often leads cryptanalyst to attacks very close to brute force ones, or requiring completely impractical amounts of chosen/known inputs up to the full codebook.

In works like [BDD+12] authors consider Low-Data Complexity attacks on reduced-rounds of AES, that is they apply attacks assuming the attacker has limited resources, e.g. few plaintext/ciphertext pairs, which is often much more relevant in practice than attacks only aiming at the highest number of rounds. The results of this work have then been improved in [BDF11]. In that paper, authors set up tools which try to find attacks automatically by searching some classes of Guess-and-Determine and Meet-in-the-Middle attacks. These tools take as input a system of equations that describes the cryptographic primitive and some constraints on the plaintext and ciphertext variables. Then, they first run a search for an "ad hoc" solver for the equations to solve, build it, and then run it to obtain the actual solutions. Other competitive low-data key-recovery attacks on 3- and 4-round AES based on 2-round truncated differential distinguishers with prob. 1 has been proposed at ToSC/FSE 2017 by Grassi *et al.* [GRR16].

Polytopic Attack. Another attack competitive in the low-data complexity scenario is the Polytopic Cryptanalysis [Tie16a], which is a generalization of differential cryptanalysis. *Polytopic cryptanalysis* has been introduced by Tiessen at Eurocrypt 2016, and it can be viewed as a generalization of standard differential cryptanalysis. Consider a set of $d \geq 2$ couples of plaintexts $(p_0, p_0 \oplus \alpha_1), (p_0, p_0 \oplus \alpha_2), ..., (p_0, p_0 \oplus \alpha_d)$ with one plaintext in common (namely p_0), called d-poly. The idea of polytopic cryptanalysis is to exploit the probability that the input set of differences $\alpha \equiv (\alpha_1, \alpha_2, ..., \alpha_d)$ is mapped into an output set of differences $\beta \equiv (\beta_1, \beta_2, ..., \beta_d)$ after r rounds. If this probability[16] - which depends on the S-Box details - is different from the corresponding probability in the case of a random permutation, it is possible to set up distinguishers or key-recovery attacks. Impossible polytopic cryptanalysis focuses on the case in which the probability of the previous event is zero. In [Tie16a], an impossible 8-polytopic is proposed for 2-round AES, which allows to set up low-data key-recovery attacks on 4- and 5-round AES.

3.3.10. Related-Key Attacks

The related-key attack model [Bih93; Bih94] is a class of cryptanalytic attacks in which the attacker knows or chooses a relation between several keys and is given access to encryption/decryption functions with all these keys. The goal of the attacker is to find the actual secret keys. In the simplest form of this attack[17], this relation between the keys is just a XOR with a constant, that is $k_2 = k_1 \oplus C$, where the constant C is chosen by the attacker. This type of relation allows the attacker to trace the propagation of XOR differences induced by the key difference C through the key schedule of the cipher.

For the AES case, related-key attacks have been proposed for full AES-256 and full AES-192 [BKN09; BK09], while no related-key attack on full AES-128 has been proposed in the literature. In both attacks on AES-192 and AES-256, authors minimize the number of active S-Boxes in the key-schedule by looking for local collisions.

[16] We remark that *the probability of polytopic trails is usually much lower than the probability of trails in differential cryptanalysis, that is simple polytopic cryptanalysis can not in general outperform standard differential cryptanalysis* - see Sect. 2 of [Tie16a] for details. For this reason, Polytopic Cryptanalysis can not be more competitive that differential cryptanalysis in the general setting, but it can outperform it when one works in the low-data scenario.

[17] We mention that more complex forms of this attack allow other (possibly non-linear) relations between the keys.

4 Subspace Trail Cryptanalysis

Invariant subspace cryptanalysis is a cryptanalytic technique that is powerful for certain block ciphers. If there exists an invariant subspace for the round function and for the key schedule, then this technique can be used to mount fast distinguishers and key recovery. However, if such symmetries do not exist or are not found, invariant subspace cryptanalysis is not applicable. This leads to the natural question: *Can subspace properties still be used, even if no special symmetries or constants allow for invariant subspaces?*

In [GRR16], we answered this question in the affirmative. While invariant subspace cryptanalysis relies on iterative subspace structures, our analysis is concerned with *trails* of different subspaces. With this more generic treatment of subspaces we do no longer rely on specific choices of round constants or subkeys, and the resulting method is as such a potentially more powerful attack vector.

Interestingly, a strong relation exists between subspace trails and (impossible) truncated differential cryptanalysis. As a result, *subspace trail turns out to be an alternative notation that can be exploited to formally describe several attacks in the literature.* While an alternative representation of a cipher can obviously be regarded in itself neither as a design nor as a cryptanalysis result, *the simplicity of a new representation of a cipher can play a significant heuristic role in the investigation of distinguishers and key-recovery attacks.*

To support this claim, we report some examples in the literature that illustrate that the choice of an appropriate description of a cipher may be very useful for highlighting some of its structural features and serve as a starting point for its cryptanalysis or for optimized implementations.

As first example, the so-called ladder representation of the Feistel scheme – which is strictly equivalent to its more traditional twisted representation for any even number of rounds – is helpful for understanding some attacks against DES and DES-like ciphers, as the Davies-Murphy attack [DM95].

In the case of AES, several alternative representations have been proposed [FSW01; MR02] to highlight some aspects of its algebraic structure. In [BB02; BCBP03] it was shown that numerous *dual ciphers of AES* – i.e. equivalent descriptions of AES up to fixed, easy to compute and to invert bijective mappings on the plaintexts, the ciphertexts, and the keys – can be obtained by applying appropriately chosen modifications to the irreducible polynomial used to represent $GF(2^8)$, the affine transformation in the S-Box, the coefficients of MixColumns, etc. While these dual ciphers can be considered as equivalent representations of AES, these representations essentially preserve the structure of the round function of the AES up to small variations on the exact parameter of each elementary transformation.

Another representation introduced by the designers of AES [DR06] is the so-called *super S-Box* (or super-Sbox) *representation* of two AES round. It allows to describe two consecutive AES rounds as the composition of one single non-linear operation, namely a range of four parallel 32-bit to 32-bit key-dependent S-Boxes and several affine transformations. This representation – useful e.g. for the analysis of AES differentials over two rounds – was subsequently re-used in [GP10; LMS+15] in order to extend the so-called rebound attacks [MRST09; LMR+09] on AES-like permutations by at least one round: this improved rebound technique, sometimes referred to as super S-Box cryptanalysis, was shown to be applicable (at least) in two related contexts, the cryptanalysis of AES-like hash functions and the investigation of so-called known-key distinguishers for AES-like block ciphers.

Finally, a novel representation - called "*untwisted representation*" - of two consecutive AES rounds that results from an extra simplification of the super S-Box representation was introduced in [Gil14]. The simplification relates to the description of the affine transformations that surround the 32-bit super S-Boxes. In there, authors show that all these affine transformations can be replaced by one simple 32-bit oriented affine transformation that operates on the rows of the 4×4 matrix of bytes representing the current state. As a result, in the untwisted representation of AES, two consecutive AES rounds are viewed as the composition of a non-linear transformation S and an affine transformation R that respectively operate on the four 32-bit columns and on the four 32-bit rows of their 128-bit input. This representation has been introduced to analyze the resistance of AES-like ciphers or AES-based hash functions against some structural attacks, and in order to present new known-key distinguishers on 8- and 10-round AES.

4.1. Subspace Trail Cryptanalysis

4.1.1. Invariant Subspace Cryptanalysis

Invariant subspace cryptanalysis [LAAZ11; LMR15] can be a powerful cryptanalytic tool. Let F denote a round function in an iterative key-alternating block cipher $E_K(\cdot)$:

$$E_K(m) = k_n \oplus F(\dots k_2 \oplus F(k_1 \oplus F(k_0 \oplus m))),$$

where the round keys $k_0, \dots, k_n$ are derived from the master key K using some key schedule f: $(k_0, \dots, k_n) = f(K)$. Assume there exists a coset[1] $V \oplus a$ such that $F(V \oplus a) = V \oplus a'$. Then if the round key K resides in $V \oplus (a \oplus a')$, it follows that

$$F_K(V \oplus a) := F(V \oplus a) \oplus K = V \oplus a$$

and we get an *iterative invariant subspace.*

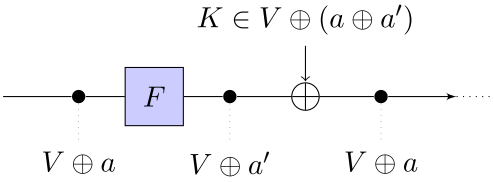


Figure 4.1.: Invariant Subspace.

A slightly more powerful property can occur if for each a, there exists unique b such that

$$F_K(V \oplus a) = F(V \oplus a) \oplus K = V \oplus b$$

meaning that the subspace property is invariant, but not the initial coset[2]. That is, for each initial coset $V \oplus a$, its image under the application of F_K is another coset of V, in general different from the initial one. Equivalently, the initial coset $V \oplus a$ is mapped into another coset $V \oplus b$, where b depends on a and on the round key.

Definition 3 (Invariant Subspace Trail). *Let K_{weak} be a set of keys. Given $k \in K_{weak}$, let $k \equiv (k^0, k^1, \dots, k^r)$ where k^j is the j-th round key. For each $k \in K_{weak}$, the subspace U generates an*

[1] Let W a vector space and V a subspace of W. If a is an element of W, a *coset* $V \oplus a$ of V in W is a subset of the form $V \oplus a = \{v \oplus a \,|\, \forall v \in V\}$. We recall that two different cosets $V \oplus a$ and $V \oplus b$ (i.e. $a \neq b$) of the same generic subspace V are *equal* if and only if $a \oplus b \in V$.

[2] Note that it is not necessary that $a = b$ in order to set up an invariant subspace attack. Indeed, remember that the round-keys are in general different, which means that they belong to different cosets of V.

invariant subspace trail *of length r for the function $F_k(\cdot) \equiv F(\cdot) \oplus k$ if for each $i = 1, \ldots, r$ there exists a non-empty set $A_i \subseteq U^C$ – where U^C is the complementary subspace of U – for which the following property holds: for each $a_i \in A_i$, there exists (unique) $a_{i+1} \in A_{i+1}$ such that*

$$F_{k^i}(U \oplus a_i) \equiv F(U \oplus a_i) \oplus k^i = U \oplus a_{i+1}.$$

Resistance against Invariant Cryptanalysis. Many lightweight block ciphers apply a very simple key schedule in which the round keys only differ by addition of a round-specific constant. Due to a poor choice of round constants, several of those schemes were recently broken using invariant attacks, e.g. PRINTcipher [LAAZ11], Robin, iSCREAM and Zorro [LMR15], Midori [GJN+16] and Haraka [Jea16a]. In [BCLR17], authors showed how to choose the round constants in order to prove resistance against invariant subspace (or more generally invariant sets) in the case of identical round keys (up to the addition of round constants).

4.1.2. Subspace Trail Cryptanalysis

A generalization of this concept is a subspace trail [GRR16]. In the simplest case, we look for pairs of subspaces V_1 and V_2 such that

$$F(V_1 \oplus a) \oplus K = V_2 \oplus b$$

holds for any constant a, that is for each a there exists unique b for which the previous equivalence is satisfied.

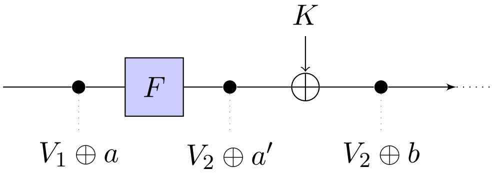


Figure 4.2.: Subspace Trail.

A *subspace trail* of length r is then simply a set of $r + 1$ subspaces $(U_1, U_2, \ldots, U_{r+1})$ that satisfy

$$F(U_i \oplus a_i) \oplus K \subseteq U_{i+1} \oplus a_{i+1}.$$

When the relation holds with equality, the trail is called a *constant-dimensional* subspace trail. In this case, if F_K^t denotes the application of t rounds with fixed keys, it follows that

$$F_K^t(U_1 \oplus a_1) = U_{t+1} \oplus a_{t+1}.$$

Definition 4 (**Subspace Trail** [GRR16])**.** *Let $(U_1, U_2, ..., U_{r+1})$ denote a set of $r+1$ subspaces with* $\dim(U_i) \leq \dim(U_{i+1})$. *If for each $i = 1, ..., r$ and for each $a_i \in U_i^C$ – where U_i^C is the complementary subspace of U_i - there exist (unique) $a_{i+1} \in W_{i+1}$ such that for each key k*

$$F_k(U_i \oplus a_i) \equiv F(U_i \oplus a_i) \oplus k \subseteq U_{i+1} \oplus a_{i+1},$$

then $(U_1, U_2, ..., U_{r+1})$ is subspace trail of length r for the function F_K. If all the previous relations hold with equality, the trail is called a constant-dimensional subspace trail.

Note that a_{i+1} depends on a_i and on the secret round key - to simplify notation we use a_{i+1} instead of $a_{i+1}(a_i, k)$. With subspace structures at hand, we might ask questions about the probability that ciphertexts or sums of ciphertexts reside in certain subspaces, given that the plaintexts obey certain subspace structure (e.g. their sum is also in a fixed subspace). If the sum is over two texts this approaches resembles (truncated) differential cryptanalysis, if the sum is over more it can resemble integral cryptanalysis.

Subspace Trail and Truncated Differential Cryptanalysis

As highlighted in [BLN14; GRR16; BLN17; LTW18], there is a strong connection between subspace trails and truncated differentials.

Let's focus for simplicity only on truncated differentials of probability 1, which can be described as affine spaces of differences.

Definition 5. *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a permutation. A truncated differential of probability one is defined by a pair of affine subspaces $(U \subseteq \mathbb{F}_2^n, s \in \mathbb{F}_2^n)$ and $(V \subseteq \mathbb{F}_2^n, t \in \mathbb{F}_2^n)$ for which:*

$$\forall \alpha \in U, \forall x \in \mathbb{F}_2^n : \qquad F(x) \oplus F(x \oplus \alpha \oplus s) \in V \oplus t.$$

Let's consider first the case $s = t = 0$. If $s = t = 0$, this simply states that each coset of U is mapped into a coset of V. Indeed, if (U, V) is a subspace trail of length 1, then

$$\forall a \in \mathbb{F}_2^n : \qquad \exists b \in \mathbb{F}_2^n \quad \text{s.t.} \quad F(U \oplus a) \subseteq V \oplus b$$

if and only if

$$\forall \alpha \in U, \forall x \in \mathbb{F}_2^n : \qquad F(x) \oplus F(x \oplus \alpha) \in V.$$

What happens if $s, t \neq 0$? In [LTW18], authors claim that a subspace trail (U, V) of length 1 (that is, $F(U \oplus a) \subseteq V \oplus b$) "*determine* [only] *a truncated differential with linear subspaces* [– that is $s = t = 0$ –] *that holds with probability one*" (see Corollary 2). In other words, subspace trail implies truncated differentials, while vice-versa is not true in general. As a result, "*while subspace trails are included in truncated differentials (as linear subspaces are a special case of affine subspaces), the converse is not true in general. In other words, using truncated differentials we obtain a bit more information on the actual structure of the investigated function.*".

In the following, we show that *a subspace trail always implies a truncated differential with affine subspaces, which means that*

subspace trail and truncated differential with affine subspaces are completely equivalent.

Lemma 1. *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a (keyed/unkeyed) permutation. Let (U, V) be a subspace trail of $F(\cdot)$, that is each coset of U (i.e., $U \oplus a$) is mapped into a coset of V (i.e., $V \oplus b$):*

$$\forall a : \qquad \exists b \text{ s.t. } F(U \oplus a) \subseteq V \oplus b.$$

Such subspace trail implies the existence of truncated differentials with prob. 1 defined by pairs of affine subspaces $(U \subseteq \mathbb{F}_2^n, s \in \mathbb{F}_2^n)$ and $(V \subseteq \mathbb{F}_2^n, t \in \mathbb{F}_2^n)$.

Proof. Since we already analyzed the case $s = t = 0$, we focus on the case $s, t \neq 0$.

Let (U, V) be a subspace trail of the (keyed/unkeyed) permutation F. By definition, $\forall a \in \mathbb{F}_2^n$, there exists $b \in \mathbb{F}_2^n$ s.t. $F(U \oplus a) \subseteq V \oplus b$. Thus, given $a_0, a_1 \in \mathbb{F}_2^n$ fixed but arbitrary s.t. $a_0 \neq a_1$ and $a_0 \oplus a_1 \notin U$ (equivalently, this second request implies that $U \oplus a_0 \neq U \oplus a_1$), there exist $b_0, b_1 \in \mathbb{F}_2^n$ s.t.

$$F(U \oplus a_0) \subseteq V \oplus b_0 \qquad \text{and} \qquad F(U \oplus a_1) \subseteq V \oplus b_1.$$

Note that if $F(U \oplus a_0) = V \oplus b_0$ and if $F(U \oplus a_1) = V \oplus b_1$, then $b_0 \oplus b_1 \notin V$. Indeed, $b_0 \oplus b_1 \in V$ implies $V \oplus b_0 = V \oplus b_1$, that is $U \oplus a_0 = U \oplus a_1$ or equivalently $a_0 \oplus a_1 \in U$, which is a contradiction. Instead in the case in which $F(U \oplus a_0) \subset V \oplus b_0$ and if $F(U \oplus a_1) \subset V \oplus b_1$, then both cases $b_0 \oplus b_1 \notin V$ and $b_0 \oplus b_1 \in V$ (i.e. $V \oplus b_0 = V \oplus b_1$) can occur.

By definition of subspace trail, this means that $\forall i = 0, 1$

$$\forall z \in U : \qquad F(z \oplus a_i) \in V \oplus b_i$$

that is

$$\forall w, z \in U : \qquad F(z \oplus a_0) \oplus F(w \oplus a_1) \in V \oplus b_0 \oplus b_1$$

since V is a subspace (that is, given $x, y \in V$, then $x \oplus y \in V$). Since the previous result is independent of the actual values of a_0 and a_1, it follows that

$$\forall \alpha \in U, \forall x \in \mathbb{F}_2^n : \qquad F(x) \oplus F(x \oplus \alpha \oplus s) \in V \oplus t$$

where $t = b_0 \oplus b_1$, $x = z \oplus a_0$, $\alpha = z \oplus w$ and $s = a_0 \oplus a_1$, which is exactly the definition given for truncated differentials. Note that if $b_0 \oplus b_1 \in V$, then the previous result holds in the same way using $t = 0$ (similar for $s = a_0 \oplus a_1$ if $a_0 \oplus a_1 \in U$).

□

A similar result can be derived also for truncated differential of probability lower than 1. To give a concrete example, assume to know two subspace trails (U, V) and (W, Z) of length respectively r and s:

$$\begin{aligned} \forall a \in \mathbb{F}_2^n : \quad \exists b \in \mathbb{F}_2^n \quad \text{s.t.} \quad F^r(U \oplus a) \subseteq V \oplus b \\ \forall a' \in \mathbb{F}_2^n : \quad \exists b' \in \mathbb{F}_2^n \quad \text{s.t.} \quad F^s(W \oplus a') \subseteq Z \oplus b'. \end{aligned}$$

It follows that[3]

$$\forall \alpha \in U, \forall x \in \mathbb{F}_2^n : \qquad F(x) \oplus F(x \oplus \alpha \oplus s) \in Z \oplus t$$

with probability $|V \cap W|/|V|$, where $|X|$ denotes the cardinality of the set X. The result follows immediately from the facts that

$$\begin{aligned} \forall \alpha \in U, \forall x \in \mathbb{F}_2^n : \qquad & F(x) \oplus F(x \oplus \alpha \oplus s) \in V \oplus t \\ \forall \alpha' \in W, \forall x \in \mathbb{F}_2^n : \qquad & F(x) \oplus F(x \oplus \alpha' \oplus s') \in Z \oplus t' \end{aligned}$$

and from the fact that

$$Prob\big[x \in W \,\big|\, x \in V\big] = \frac{|V \cap W|}{|V|}.$$

Similarly

$$\forall \alpha \in W, \forall x \in \mathbb{F}_2^n : \qquad F(x) \oplus F(x \oplus \alpha \oplus s) \in V \oplus t$$

with probability $|Z \cap U|/|Z|$.

Concrete examples are given in the following for 3 and 4 rounds of AES.

4.1.3. Weak-Key Subspace Trails

Invariant *versus* Subspace Trail

One may ask what relations hold between the "invariant subspace trail" definition and the "subspace trail" one. Here we highlight two obvious but important differences among them.

First, subspace trails are clearly much more general as they allow different spaces in the domain and the co-domain of $F(\cdot)$.

Second, subspace trails are by far more restrictive, as not only one coset of the subspace has to be mapped to one coset of (a potentially different) subspace, but rather all cosets have to be mapped to cosets. In more details, observe that a subspace trail for F will extend to a subspace trail for E_k for any choice of round keys. This is a simple consequence as, if $F(V_i \oplus a_i) \subseteq V_{i+1} \oplus a_{i+1}$ then $F(V_i \oplus a_i) \oplus k_i \subseteq V_{i+1} \oplus a'_{i+1}$ for a suitable a'_{i+1}. In other words, the key addition changes only the coset of the subspace U_{i+1}, while it does not affect the subspace itself.

[3] We emphasize that no assumption is made about the relation between U and Z or/and between V and W.

> *Not only do subspace trails work for all keys, they are also completely independent of the key schedule.* Here, invariant subspace attacks behave very differently. In strong contrast, *invariant subspace attacks are always weak-key attacks by nature.*

Namely, again focusing on the key-alternating cipher from above, in order to extend the invariant subspace $V \oplus a_i \mapsto V \oplus a_{i+1}$ to the whole cipher, we need all round keys to be in a specific coset of V namely, $k_i \in V \oplus (a_{i+1} \oplus b_i)$ (where $F(V \oplus a_i) = V \oplus b_i$). If this is fulfilled, then clearly $F_k(V \oplus a_i) = F(V \oplus a_i) \oplus k = V \oplus b_i \oplus k = V \oplus a_{i+1}$ which then is iterative for any number of rounds. As all round keys have to fulfill the same condition, which can be described by a system of affine equations, the class of weak-master keys is largest in the case where all round keys are actually identical to the master key itself.

Weak-Key Subspace Trails

Using the previous discussion as starting point, here we introduce the "weak-key subspace trails". The key idea is to stick to the property of invariant subspace attacks where only one coset of a subspace is mapped to one coset of a subspace. However, borrowing from subspace trails, we allow those subspaces to be different for each round. As this will again restrict the choice of round keys that will keep this property invariant to a class of weak-keys we call this combination weak-key subspace trails (or simply, weak subspace trails). The formal definition is the following.

Definition 6 (**Weak-Key Subspace Trail** [GLR+18])**.** *Let K_{weak} be a set of keys. Given $k \in K_{weak}$, let $k \equiv (k^0, k^1, ..., k^r)$ where k^j is the j-th round key. Let $(U_1, U_2, \ldots, U_{r+1})$ denote a set of $r + 1$ subspaces with $\dim(U_i) \leq \dim(U_{i+1})$. For each $k \in K_{weak}$, $(U_1, U_2, \ldots, U_{r+1})$ is a* weak subspace trail *of length r for the function $F_k(\cdot) \equiv F(\cdot) \oplus k$ if for each $i = 1, \ldots, r$ there exists a non-empty set $A_i \subseteq U_i^C$ – where U_i^C is the complementary subspace of U_i – for which the following property holds: for each $a_i \in A_i$, there exists (unique) $a_{i+1} \in A_{i+1}$ such that*

$$F_{k^i}(U_i \oplus a_i) \equiv F(U_i \oplus a_i) \oplus k^i \subseteq U_{i+1} \oplus a_{i+1}.$$

If all the previous relations hold with equality, the trail is called a weak constant-dimensional subspace trail*.*

Usually, the set $A_i \subseteq W_i$ reduces to a single element a_i, that is $A_i \equiv \{a_i\}$. Moreover, we emphasize that:

- if K_{weak} is equal to the whole set of keys and if $A_i = W_i$ for each $i = 0, ..., r + 1$, then previous definition reduces/corresponds to Def. 4 for subspace trail;
- if $U_i = U_{i+1}$ for each i, then previous definition reduces/corresponds to Def. 3 for invariant subspace trail.

Clearly, this allows greater freedom for an attacker[4]. In comparison to invariant subspace attacks, *weak-key subspace trails have the potential of being better applicable to block ciphers with a non trivial key schedule.*

4.2. Subspace Cryptanalysis for AES

For a vector space V and a function F on $\mathbb{F}_{2^8}^{4\times 4}$, let $F(V) = \{F(v) \,|\, v \in V\}$ (as usual). For a subset $I \subseteq \{1, 2, \ldots, n\}$ and a subset of vector spaces $\{G_1, G_2, \ldots, G_n\}$, we define G_I as $G_I := \bigoplus_{i \in I} G_i$. We denote with $E = \{e_{0,0}, ..., e_{3,3}\}$ the unit vectors of $\mathbb{F}_{2^8}^{4\times 4}$ (e.g. $e_{i,j}$ has a single 1 in row i and column j).

[4] ***Remark.*** *In [GLR+18] we also propose an algorithm in order to (automatically) detect weak-key subspace trails. Since I did not work on such result – it was done by Friedrich Wiemer, I limit myself to refer to [GLR+18, Sect. 2.4] for all details.*

4.2.1. Subspaces for AES

In the following we define four families of subspaces essential to AES: the diagonal spaces $\mathcal{D}_I$, the inverse-diagonal spaces $\mathcal{ID}_I$, the column spaces $\mathcal{C}_I$ and the mixed spaces $\mathcal{M}_I$.

Definition 7 (**Column Spaces** [GRR16])**.** *The column spaces $\mathcal{C}_i$ are defined as*

$$\mathcal{C}_i = \langle e_{0,i}, e_{1,i}, e_{2,i}, e_{3,i} \rangle.$$

For instance, the column space $\mathcal{C}_0$ corresponds to the symbolic matrix

$$\mathcal{C}_0 = \left\{ \begin{bmatrix} x_1 & 0 & 0 & 0 \\ x_2 & 0 & 0 & 0 \\ x_3 & 0 & 0 & 0 \\ x_4 & 0 & 0 & 0 \end{bmatrix} \middle| \forall x_1, x_2, x_3, x_4 \in \mathbb{F}_{2^8} \right\}.$$

Definition 8 (**Diagonal Spaces** [GRR16])**.** *The diagonal spaces $\mathcal{D}_i$ are defined as*

$$\mathcal{D}_i = SR^{-1}(\mathcal{C}_i) = \langle e_{0,i}, e_{1,i+1}, e_{2,i+2}, e_{3,i+3} \rangle$$

where the index $i + j$ is computed modulo 4. For instance, the diagonal space $\mathcal{D}_0$ corresponds to the symbolic matrix

$$\mathcal{D}_0 = \left\{ \begin{bmatrix} x_1 & 0 & 0 & 0 \\ 0 & x_2 & 0 & 0 \\ 0 & 0 & x_3 & 0 \\ 0 & 0 & 0 & x_4 \end{bmatrix} \middle| \forall x_1, x_2, x_3, x_4 \in \mathbb{F}_{2^8} \right\}.$$

Definition 9 (**Inverse-Diagonal Spaces** [GRR16])**.** *The inverse-diagonal spaces $\mathcal{ID}_i$ are defined as*

$$\mathcal{ID}_i = SR(\mathcal{C}_i) = \langle e_{0,i}, e_{1,i-1}, e_{2,i-2}, e_{3,i-3} \rangle.$$

where the index $i - j$ is computed modulo 4. For instance, $\mathcal{ID}_0 = SR(\mathcal{C}_0)$ corresponds to the symbolic matrix

$$\mathcal{ID}_0 = \left\{ \begin{bmatrix} x_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 \\ 0 & 0 & x_3 & 0 \\ 0 & x_4 & 0 & 0 \end{bmatrix} \middle| \forall x_1, x_2, x_3, x_4 \in \mathbb{F}_{2^8} \right\}.$$

The last type of subspaces we define are called mixed subspaces.

Definition 10. *[**Mixed spaces** [GRR16]] The i-th mixed subspace $\mathcal{M}_i$ is defined as*

$$\mathcal{M}_i = MC(\mathcal{ID}_i).$$

These subspaces are formed by applying ShiftRows and then MixColumns to a column space. For instance, $\mathcal{M}_0$ corresponds to symbolic matrix

$$\mathcal{M}_0 = \left\{ \begin{bmatrix} \text{0x02} \cdot x_1 & x_4 & x_3 & \text{0x03} \cdot x_2 \\ x_1 & x_4 & \text{0x03} \cdot x_3 & \text{0x02} \cdot x_2 \\ x_1 & \text{0x03} \cdot x_4 & \text{0x02} \cdot x_3 & x_2 \\ \text{0x03} \cdot x_1 & \text{0x02} \cdot x_4 & x_3 & x_2 \end{bmatrix} \middle| \forall x_1, x_2, x_3, x_4 \in \mathbb{F}_{2^8} \right\}.$$

The essential subspaces in AES are built from diagonal spaces $\mathcal{D}_i$, inverse-diagonal spaces $\mathcal{ID}_i$, column spaces $\mathcal{C}_j$ and mixed spaces $\mathcal{M}_k$. There are four of each of these spaces, and direct sums of them result in higher-dimensional diagonal, inverse-diagonal, column and mixed spaces.

Definition 11 ([GRR16])**.** *Given $I \subseteq \{0,1,2,3\}$ where $0 < |I| \leq 3$, we define:*

$$\mathcal{C}_I = \bigoplus_{i\in I} \mathcal{C}_i, \qquad \mathcal{D}_I = \bigoplus_{i\in I} \mathcal{D}_i, \qquad \mathcal{ID}_I = \bigoplus_{i\in I} \mathcal{ID}_i \qquad \mathcal{M}_I = \bigoplus_{i\in I} \mathcal{M}_i.$$

The dimension[5] of any of the spaces $\mathcal{D}_I, \mathcal{ID}_I, \mathcal{C}_I$ and $\mathcal{M}_I$ is $4 \cdot |I|$. Before going on, we remark that the complements of $\mathcal{D}_I, \mathcal{C}_I, \mathcal{ID}_I, \mathcal{M}_I$ are simply the (respective) orthogonal $\mathcal{D}_I^\perp, \mathcal{C}_I^\perp, \mathcal{ID}_I^\perp, \mathcal{M}_I^\perp$. This follows immediately by the fact that[6]

$$\mathcal{D}_I \oplus \mathcal{D}_I^\perp = \mathcal{C}_I \oplus \mathcal{C}_I^\perp = \mathcal{ID}_I \oplus \mathcal{ID}_I^\perp = \mathcal{M}_I \oplus \mathcal{M}_I^\perp = \mathbb{F}_{2^8}^{4\times 4}$$

for each $I \subseteq \{0,1,2,3\}$.

4.2.2. Subspace Trails of AES

Here we prove that $\{\mathcal{D}_I, \mathcal{C}_I, \mathcal{M}_I\}$ is a subspace trail of AES of length 2. To do this, we show that $\{\mathcal{D}_I, \mathcal{C}_I\}$ and $\{\mathcal{C}_I, \mathcal{M}_I\}$ are two subspace trails of AES of length 1. The result follows immediately.

Subspace Trail: $\{\mathcal{D}_\mathbf{I}, \mathcal{C}_\mathbf{I}\}$. It is easy to see that SubBytes maps cosets of diagonal and column spaces to cosets of diagonal and column spaces. Since SubBytes operates on each byte individually and it is bijective, and since the bytes of column and diagonal spaces are independent, its only effect is to change the coset. It is also easy to see that ShiftRows maps a coset of a diagonal space to a coset of a column space, since diagonals are mapped to columns, and it maps a coset of a column space to a coset of an inverse-diagonal space. The effect of MixColumns to a columns space $\mathcal{C}_I \oplus a$ is simply to change the coset, since applying the MixColumns matrix to a column space $\mathcal{C}_i$ has no effect.

Lemma 2 ([GRR16])**.** *Let $I \subseteq \{0,1,2,3\}$ where $0 < |I| \leq 3$ and $a \in \mathcal{D}_I^\perp$. There exists unique $b \in \mathcal{C}_I^\perp$ such that*

$$R_K(\mathcal{D}_I \oplus a) = \mathcal{C}_I \oplus b.$$

Proof. As we have just seen, since SubBytes is bijective and operates on each byte independently, it simply changes the coset $\mathcal{D}_I \oplus a$ to $\mathcal{D}_I \oplus a'$, where $a'_{i,j} = \text{S-Box}(a_{i,j})$ for each $i, j = 0, ..., 3$. ShiftRows simply moves the bytes of $\mathcal{D}_I \oplus a'$ to a column space $\mathcal{C}_I \oplus b^{'}$, where $b' = SR(a')$. MixColumns affects only the constant columns, thus $MC(\mathcal{C}_I \oplus b^{'}) = \mathcal{C}_I \oplus MC(b') = \mathcal{C}_I \oplus b^{''}$. Key addition then changes the coset to $\mathcal{C}_I \oplus b$. □

This simply states that a coset of a sum of diagonal spaces $\mathcal{D}_I$ encrypt to a coset of a corresponding sum of column spaces $\mathcal{C}_I$ through one round.

Subspace Trail: $\{\mathcal{C}_\mathbf{I}, \mathcal{M}_\mathbf{I}\}$. Similarly to before, a coset of a sum of column spaces $\mathcal{C}_I$ encrypts to a coset of the corresponding sum of mixed spaces $\mathcal{M}_I$ over one round.

Lemma 3 ([GRR16])**.** *Let $I \subseteq \{0,1,2,3\}$ where $0 < |I| \leq 3$ and $a \in \mathcal{C}_I^\perp$. There exists unique $b \in \mathcal{M}_I^\perp$ such that*

$$R_K(\mathcal{C}_I \oplus a) = \mathcal{M}_I \oplus b.$$

[5] Since we are working over the field $\mathbb{F}_{2^8}$, we consider the dimension of the subspace as the number of active and independent bytes. As a result, the dimension of the subspaces is constant through SubBytes and MixColumns operations.

[6] The equivalence $\mathcal{M}_I \oplus \mathcal{M}_I^\perp = \mathbb{F}_{2^8}^{4\times 4}$ can be obtained by $\mathcal{ID}_I \oplus \mathcal{ID}_I^\perp = \mathbb{F}_{2^8}^{4\times 4}$ by applying $MC(\cdot)$ on both sides - remember that $MC(\cdot)$ is linear.

Proof. By Def. 10, the mixed spaces $\mathcal{M}_I$ are defined as the application of the MixColumns operation to inverse-diagonal space $\mathcal{ID}_I$. Since a ShiftRows operation maps a column space to an inverse-diagonal space, a mixed space $\mathcal{M}_I$ is equivalently defined as the application of the linear layer in AES to column spaces $\mathcal{C}_I$. Since the SubBytes layer only moves a coset $\mathcal{C}_I \oplus a$ to a coset $\mathcal{C}_I \oplus a'$, it follows that for any fixed coset $\mathcal{C}_I \oplus a$, there exists $b \in \mathcal{M}_I^{\perp}$ such that $MC \circ SR \circ \text{S-Box}(\mathcal{C}_I \oplus a) \oplus K = \mathcal{M}_I \oplus b$, where $b = MC \circ SR(a') \oplus K$ and $a'_{i,j} = \text{S-Box}(a_{i,j})$ for each $i, j = 0, ..., 3$. □

Subspace Trail: $\{\mathcal{D}_\mathbf{I}, \mathcal{C}_\mathbf{I}, \mathcal{M}_\mathbf{I}\}$. Finally, we are able to prove the desired result.

Theorem 3 ([GRR16])**.** *Let $I \subseteq \{0,1,2,3\}$ where $0 < |I| \leq 3$ and $a \in \mathcal{D}_I^{\perp}$. There exists unique $c \in \mathcal{M}_I^{\perp}$ such that*

$$R_K^2(\mathcal{D}_I \oplus a) = \mathcal{M}_I \oplus c.$$

This simply states that each coset of $\mathcal{D}_I$ is mapped into a coset of $\mathcal{M}_I$ after 2 rounds, independently of the details of the S-Box and of the secret key.

4.2.3. Intersecting AES Subspaces

We continue with useful properties of AES subspaces. In this section we show the following: diagonal spaces and column spaces have non-trivial intersection, column spaces and mixed spaces have non-trivial intersection, but diagonal spaces and mixed spaces have only trivial intersection. This will be useful for creating subspace trails covering a higher number of rounds. In the following, let $I, J \subseteq \{0,1,2,3\}$ and we assume that all the indexes are taken modulo 4.

Proposition 2 ([GRR16])**.** *$\mathcal{D}_i \cap \mathcal{C}_j = \langle e_{i+j,j} \rangle$ and $\mathcal{ID}_i \cap \mathcal{C}_j = \langle e_{i-j,j} \rangle$.*

Proof. $\mathcal{D}_i$ space corresponds to a symbolic matrix with variables along the i-th diagonal, while $\mathcal{C}_j$ has variables in the j-th column. Any diagonal and column meets in exactly one byte, precisely in row $j+i$ and column j. The proof is equivalent for the intersection $\mathcal{ID}_i \cap \mathcal{C}_j$. □

It follows that $\mathcal{D}_I \cap \mathcal{C}_J = \langle e_{j+i,j} \,|\, i \in I, j \in J \rangle$ and $\mathcal{ID}_I \cap \mathcal{C}_J = \langle e_{i-j,j} \,|\, i \in I, j \in J \rangle$ ($j+i$ and $i-j$ are taken modulo 4), where the intersections have dimension $|I| \cdot |J|$.

Proposition 3 ([GRR16])**.** *$\mathcal{C}_i \cap \mathcal{M}_j = \langle MC(e_{j+i,i}) \rangle$.*

Proof. We have that $MC \circ SR(\mathcal{D}_i) = \mathcal{C}_i$ and by Def. 10, $\mathcal{M}_i = MC(\mathcal{ID}_i) = MC \circ SR(\mathcal{C}_i)$. By Lemma 2, $\mathcal{D}_i \cap \mathcal{C}_j = \langle e_{j+i,j} \rangle$. Thus it follows that $\langle MC(e_{j+i,j}) \rangle = MC \circ SR(\mathcal{D}_i) \cap MC \circ SR(\mathcal{C}_j) = \mathcal{D}_i \cap \mathcal{M}_j$. Finally, since $SR(e_{r,c}) = e_{r,c-r}$, we obtain that $\langle MC \circ SR(e_{j+i,j}) \rangle = \langle MC(e_{j+i,i}) \rangle$. □

It follows that $\mathcal{C}_I \cap \mathcal{M}_J = \langle MC(e_{j+i,i}) \,|\, i \in I, j \in J \rangle$ ($i+j$ is taken modulo 4), which has dimension $|I| \cdot |J|$.

While the spaces $\mathcal{D}_I$ and $\mathcal{C}_J$, $\mathcal{ID}_I$ and $\mathcal{C}_J$, and $\mathcal{C}_I$ and $\mathcal{M}_J$ intersect non-trivially, the spaces $\mathcal{D}_I$ and $\mathcal{M}_J$ and the spaces $\mathcal{ID}_I$ and $\mathcal{M}_J$ intersect trivially. In particular:

Proposition 4 ([GRR16])**.** *$\mathcal{D}_I \cap \mathcal{M}_J = \mathcal{ID}_I \cap \mathcal{M}_J = \{0\}$ for all I and J such that $|I| + |J| \leq 4$.*

Proof. To prove this proposition, we first consider the case $|I| = |J| = 1$, and we prove the following result.

$$\forall i, j \in \{0,1,2,3\} : \qquad \mathcal{D}_i \cap \mathcal{M}_j = \mathcal{ID}_i \cap \mathcal{M}_j = \{0\}.$$

The proof works as follows. A basis for $\mathcal{M}_j$ is given by:

$$\mathcal{M}_j = \langle MC(e_{0,j}), MC(e_{1,j-1}), MC(e_{2,j-2}), MC(e_{3,j-3}) \rangle,$$

while a basis for $\mathcal{D}_i$ is given by $\mathcal{D}_i = \langle\langle e_{0,i}, e_{1,i+1}, e_{2,i+2}, e_{3,i+3} \rangle$, where in both cases the indexes are taken modulo 4.

Suppose by contradiction that $\mathcal{D}_i$ and $\mathcal{M}_j$ has a nonzero intersection. This implies that there exist x_k and y_k for $k = 0, ..., 3$ such that

$$\bigoplus_{k=0}^{3} [x_{k-i} \cdot e_{k-i,k} \oplus y_{k+j} \cdot MC(e_{k+j,k})] = 0. \tag{4.1}$$

has a nontrivial solution (where at least one x_k or/and y_k is different from zero). The only possible solution of the previous equivalence is given by

$$x_{k-i} \cdot e_{k-i,k} \oplus y_{k+j} \cdot MC(e_{k+j,k}) = 0$$

for each k (note that $e_{\cdot,n}$ and $e_{\cdot,m}$ lie on different columns for $n \neq m$ - similar for $MC(e_{\cdot,n})$ and $MC(e_{\cdot,m})$). This is clearly impossible since $e_{k-i,k}$ and $MC(e_{k+j,k})$ are linearly independent for each $k = 0, ..., 3$. Thus, $\mathcal{D}_i$ and $\mathcal{M}_j$ intersect only in zero.

Coming back to the generic case $I, J \subseteq \{0, 1, 2, 3\}$, as long as $|I| + |J| \leq 4$, we have that any combinations of subspaces $\mathcal{D}_I$ and $\mathcal{M}_J$ only intersect in the zero vector. Indeed, consider the sum over k defined in Eq. (4.1). If $|I| + |J| \leq 4$, then for each k (i.e. for each column) there are at most four terms. Among them, there is at least one term of the form $\langle e_{\cdot,k} \rangle$ and at least one of the form $\langle MC(e_{\cdot,k}) \rangle$. Thus, equation (4.1) has only trivial solutions. Instead, note that this is not true if $|I| + |J| > 4$. Indeed, in this case for each k (i.e. for each column), the equation (4.1) has at least 5 terms. Since there are only 4 rows, it is always possible to find non trivial solutions.

The proof is equivalent for the intersection $\mathcal{ID}_I \cap \mathcal{M}_J$. □

4.3. Truncated Distinguishers for AES

In this section, we show that the subspace trail notation is a valid notation in order to describe truncated (and impossible) distinguishers for up to 4-round AES. In other words, the "classical" truncated differential notation and the subspace trail one are basically equivalent. This is due to the fact that *the difference of two texts t^1 and t^2 can be described by the fact that t^1 and t^2 belong to the same coset of a particular subspace $\mathcal{X}$, that is $t^1 \oplus t^2 \in \mathcal{X}$.*

For concrete examples, consider the following. If two texts t^1 and t^2 are equal except for the bytes in the i-th diagonal[7] for each $i \in I$, then they belong to the same coset of $\mathcal{D}_I$. A coset of $\mathcal{D}_I$ corresponds to a set of $2^{32 \cdot |I|}$ texts with $|I|$ active diagonals. Again, two texts t^1 and t^2 belong to the same coset of $\mathcal{ID}_I$ if the difference of the bytes that lie in the i-th anti-diagonal for each $i \notin I$ is equal to zero. Similar considerations hold for the column space $\mathcal{C}_I$ and the mixed space $\mathcal{M}_I$.

4.3.1. Truncated Differential for 2-round AES

As we have seen, diagonal spaces are encrypted over two rounds to ciphertexts in mixed subspaces. More formally, for each $a \in \mathcal{D}_I^{\perp}$, there exists unique $b \in \mathcal{M}_I^{\perp}$ such that $R^{(2)}(\mathcal{D}_I \oplus a) = \mathcal{M}_I \oplus b$.

Consequently, we get the following properties. If two plaintexts belong to the same coset of a diagonal space $\mathcal{D}_I$, then their 2-round encryptions belong to the same coset of a mixed space $\mathcal{M}_I$. In particular, for a two round encryption R^2 with fixed keys, we have that

$$Prob\big[R^2(p^1) \oplus R^2(p^2) \in \mathcal{M}_I \,|\, p^1 \oplus p^2 \in \mathcal{D}_I\big] = 1 \tag{4.2}$$

for nonzero set I of $\{0, 1, 2, 3\}$ (i.e. $|I| \neq 0$).

[7] The i-th diagonal of a 4×4 matrix A is defined as the elements that lie on row r and column c such that $r - c = i \bmod 4$. The i-th anti-diagonal of a 4×4 matrix A is defined as the elements that lie on row r and column c such that $r + c = i \bmod 4$.

Since for a random permutation $\Pi(\cdot)$ it holds that

$$Prob\big[\Pi(p^1) \oplus \Pi(p^2) \in \mathcal{M}_I \,|\, p^1 \oplus p^2 \in \mathcal{D}_I\big]= (2^{-32})^{4-|I|}, \tag{4.3}$$

it is possible to exploit this property to distinguish two rounds of AES from a random permutation. In particular, the difference between the two probabilities is maximized by choosing $|I| = 1$. In this last case, two pairs of plaintexts/ciphertexts (p^1, c^1) and (p^2, c^2) where $p^1 \oplus p^2 \in \mathcal{D}_I$ are largely sufficient to distinguish the two permutations.

4.3.2. Truncated Differential for 3-round AES

While no deterministic subspace trail can be set up for 3-round AES, we can exploit the subspace trail notation to easily describe any truncated differential on 3-round AES.

Consider a coset of $\mathcal{D}_I$ as starting point. After one round, this coset is mapped into a coset of $\mathcal{C}_I$ with probability 1 - see Lemma 2. Thus, if we consider two elements that belong to the same cosets of $\mathcal{D}_I$, after one round they belong in the same coset of $\mathcal{C}_I$ for sure. However, at the same time and with a certain probability, it is possible that these two elements belong to the same coset of $\mathcal{C}_I \cap \mathcal{D}_J \subseteq \mathcal{D}_J$ for a certain J after one round. This happens with a certain probability, and this is the starting point for our desired result. In particular, the following proposition holds:

Proposition 5 ([GRR16]). *For any $\mathcal{C}_I$ and $\mathcal{D}_J$, we have that*

$$Prob\big[x \in \mathcal{D}_J \,|\, x \in \mathcal{C}_I\big]= (2^8)^{-4|I|+|I|\cdot|J|}. \tag{4.4}$$

That is, given a texts in $\mathcal{C}_I$, it also belongs in $\mathcal{D}_J$ with probability $(2^8)^{-4|I|+|I|\cdot|J|}$. This follows immediately by the intersection $\mathcal{C}_I \cap \mathcal{D}_J$, as shown in Prop. 2.

This result is the starting point for any truncated differential distinguisher on 3-round AES. If two plaintexts belong to the same coset of a diagonal space $\mathcal{D}_I$, then their 3-round encryption belongs to the same coset of a mixed space $\mathcal{M}_J$ with prob. $(2^8)^{-4|I|+|I|\cdot|J|}$. In particular, for a three round encryption R^3 with fixed keys, we have that

$$Prob\big[R^3(p^1) \oplus R^3(p^2) \in \mathcal{M}_J \,|\, p^1 \oplus p^2 \in \mathcal{D}_I\big]= (2^8)^{-4|I|+|I|\cdot|J|} \tag{4.5}$$

for nonzero set I of $\{0, 1, 2, 3\}$ (i.e. $|I| \neq 0$). To get the result, note that

$$\mathcal{D}_I \oplus a \xrightarrow[\text{prob. } (2^8)^{-4|I|+|I|\cdot|J|}]{R(\cdot)} \big(\mathcal{C}_I \cap \mathcal{D}_J\big) \oplus b \xrightarrow[\text{prob. } 1]{R^2(\cdot)} \mathcal{M}_J \oplus c.$$

Since for a random permutation $\Pi(\cdot)$ it holds that $Prob\big[\Pi(p^1) \oplus \Pi(p^2) \in \mathcal{M}_J \,|\, p^1 \oplus p^2 \in \mathcal{D}_I\big]= (2^{-32})^{4-|J|}$, it is possible to distinguish three rounds of AES from a random permutation. In particular, the difference between the two probabilities is maximized by choosing $|I| = 1$ and $|J| = 3$. For this choice, it turns out that the probability for 3-round AES is 2^{-8} while for a random permutation is 2^{-32}.

In this last case, 40 pairs of plaintext/ciphertext (p^i, c^i) for $1 \leq i \leq 20$ where $p^i \oplus p^j \in \mathcal{D}_I$ for each i, j are largely sufficient to distinguish the two permutations[8]. Indeed, given 40 texts, it is possible to construct $\binom{40}{2} = 780$ different couples of two pairs of plaintext/ciphertext. As a result, we expect approximately $780 \cdot 2^{-8} \simeq 3$ collisions for the AES case and $780 \cdot 2^{-32} \simeq 0$ for the random case.

Moreover, in the following we show that also *the variance[9] can be exploited in order to distinguish 3-round AES from a random permutation*, besides the mean. To the best of our knowledge, this is the first time that such consideration is explicitly made.

[8] We emphasize that 20 pairs of plaintext/ciphertext are sufficient to distinguish the two cases if the index $J \subseteq \{0, 1, 2, 3\}$ with $|J| = 3$ of the final subspace $\mathcal{M}_J$ is not fixed.

[9] Potentially, also the skewness can be used in order to set up a distinguisher.

Description in the Literature. To have a concrete comparison of the description of such distinguisher using the "classical" truncated differential notation, we recall its description proposed in [BK07]: "*a differential which starts with four active S-boxes at the* 1*st round. We choose those active S-boxes to appear in positions which arrive in one column after the ShiftRows transformation. Then with probability* 2^{-6} *four active S-boxes will collapse to three (one byte out of four getting a zero difference). After the second round the three active bytes are expanded into 12 active bytes and there will still remain 4 passive bytes. This differential can be schematically described as* $4 \to 3 \to 12$." For comparison, our notation allows to formally collect all possible cases.

4.3.3. (Impossible) Truncated Differential for 4-round AES

From now on, we assume that I and J satisfy the condition $0 < |I| + |J| \leq 4$ (which allows us to use Lemma 4).

To set up the 4-round impossible differential distinguisher, the idea is to combine two 2-round differential ones with prob. 1 such that they collapse in the middle. In particular, remember that

$$Prob\big[R^2(p^1) \oplus R^2(p^2) \in \mathcal{M}_I \,|\, p^1 \oplus p^2 \in \mathcal{D}_I\big] = 1$$

and that

$$Prob\big[x \in \mathcal{D}_J \,|\, x \in \mathcal{M}_I\big] = 0$$

if $x \neq 0$. Combining these two probabilities for 2-round yields a 4-round probability

$$Prob\big[R^4(p^1) \oplus R^4(p^2) \in \mathcal{M}_J \,|\, p^1 \oplus p^2 \in \mathcal{D}_I\big] = 0 \tag{4.6}$$

where $p^1 \neq p^2$ and $0 < |I| + |J| \leq 4$.

Since for a random permutation $\Pi(\cdot)$ it holds that $Prob\big[\Pi(p^1) \oplus \Pi(p^2) \in \mathcal{M}_J \,|\, p^1 \oplus p^2 \in \mathcal{D}_I\big] = (2^{-32})^{4-|J|}$, it is possible to distinguish four rounds of AES from a random permutation. In particular, the difference between the two probabilities is maximized by choosing $|I| = 1$ and $|J| = 3$. In this last case, $2^{17.25}$ pairs of plaintext/ciphertext (p^i, c^i) for $1 \leq i \leq 2^{17.25}$ where $p^i \oplus p^j \in \mathcal{D}_I$ for each i, j are largely sufficient to distinguish the two permutations[10].

Description in the Literature. To have a concrete comparison of the description of such distinguisher using the "classical" truncated differential notation, we recall its description proposed in [BK01]: "*If a pair of plaintexts differ by only one byte then the ciphertexts cannot be equal in any of the following combinations of bytes: (1,6,11,16), (2,7,12,13), (3,8,9,14), nor (4,5,10,15). [...] The reason is that the difference before the first MixColumn is in one byte, so after it there is difference in one column, and then after the second MixColumn the data differs in all the bytes. On the other hand, if the ciphertexts are equal in one of the four prohibited combinations of bytes then after the third MixColumn the data is equal in one column, and thus before the MixColumn the data in this column is also equal. Therefore, after the second MixColumn there are 4 bytes in which the data is equal. This is a contradiction since we showed that all the bytes of the data differ after that MixColumn. This property is indeed impossible.*" For comparison, our notation allows to formally collect all possible cases.

4.4. Weak-Key Invariant Subspace and Subspace Trails for AES

Here we show that it is possible to extend the previous results on an higher number of rounds in the case of *weak keys of AES*.

[10] We emphasize that $2^{16.25}$ pairs of plaintext/ciphertext are sufficient to distinguish the two cases if the index $J \subseteq \{0, 1, 2, 3\}$ with $|J| = 3$ of the final subspace $\mathcal{M}_J$ is not fixed.

To do this, we make use of the "weak" subspace trail notation previously defined. First of all, we define an invariant subspace $\mathcal{IS}$ and a class of weak keys of AES.

Let the subspace $\mathcal{IS}$ be defined as

$$\mathcal{IS} = \left\{ \begin{bmatrix} a & b & a & b \\ c & d & c & d \\ e & f & e & f \\ g & h & g & h \end{bmatrix} \middle| \forall a, b, c, d, \ldots, h \in \mathbb{F}_{2^8} \right\} \tag{4.7}$$

This subspace - already presented and used in e.g. [LSWD04; GNPW13; CFG+17] - is invariant under a key-less round $R(\cdot) = MC \circ SR \circ$ S-Box$(\cdot)$, since

$$\text{S-Box}(\mathcal{IS}) = \mathcal{IS} \qquad SR(\mathcal{IS}) = \mathcal{IS} \qquad MC(\mathcal{IS}) = \mathcal{IS}.$$

AES Key-Schedule. As we are going to show, *the possibility to set up a weak invariant subspace trail depends on the concrete value of the secret key and of the key schedule details.*

The problem to design a strong key schedule has been largely studied and discussed in the literature. *Usually, the target that a key schedule must satisfy is resistance against related-key attacks, while the problem of weak-keys is in general less considered. However, presence of weak-keys can have a devastating effect on the security of a cipher.*

For this reason, in the following we consider several AES key-schedules present in the literature, and for each one of them we discuss the possibility to set up a weak invariant subspace trail. In more details, we consider three categories of key schedule:

- the simplest key schedule is given by *identical subkeys* or by subkeys defined as the XOR of the whitening key and round constants - this category has been largely studied in [BCLR17], recently published at Crypto 2017;
- another category of key schedule is given by (linear) *permutation* of the byte positions: each subkey is the result of a particular permutation applied to the whitening key - e.g. the key schedule recently proposed at ToSC/FSE 2018 [KLPS17];
- finally, we consider the *AES key-schedule*[11].

For each case, we present a set of weak-keys for which the invariant subspace trail - of length equal or bigger than two - can be set up. To do this, our strategy is simply to look for keys that *(1st)* belong to the invariant subspace $\mathcal{IS}$ and *(2nd)* for which the "next round sub-key" generated by the key schedule belongs to the invariant subspace $\mathcal{IS}$. In other words, in order to find weak-keys, we initially focus on a set of 2^{64} keys - denoted by K_{weak} - "equal" to the subspace $\mathcal{IS}$ just defined, and among them we identify the keys that satisfy the second requirement just given.

4.4.1. Identical Round Keys and Weak Round Constants

The simplest possible key schedule (mainly used for lightweight ciphers) is probably obtained as follows: the r-th round subkey $k[r]$ is simply given by the XOR of the whitening key K and a round constant $RC[r]$, that is $k[r] = K \oplus RC[r]$.

[11] In [GLR+18], we also considered the key-schedule proposed by Nikolic [Nik10] at SAC 2010. This variant is obtained by introducing a small change in the current AES key schedule, which allows to improve the security against related-key attacks. In short, for obtaining each column of the new subkey, the new key schedule always uses rotation by one byte up of the previous subkey column, while AES uses a rotation only when obtaining the subkey column with an index multiple of N_k ($N_k = 4, 6, 8$ for AES-128,-192,-256).

As we show in [GLR+18], even if this change improves the security against related-key attack, it does not improve the security against weak-key attacks w.r.t. the original AES key schedule.

Consider the subspace $\mathcal{IS}$ previously defined. If for each round r the subkey $K \oplus RC[r]$ belongs to this subspace, then it is possible to set up a weak invariant subspace trail for a set of weak-keys for an arbitrary number of rounds. In particular, if $k[r] \in \mathcal{IS}$ then

$$\mathcal{IS} \xrightarrow{MC \circ SR \circ \text{S-Box}(\cdot)} \mathcal{IS} \xrightarrow{\cdot \oplus k[r]} \mathcal{IS} \tag{4.8}$$

This property, and similar symmetries in the AES round transformation, are folklore.

Clearly, with a proper choice of round constants, such properties can be easily avoided. As already mention, [BCLR17] show how to check that at least invariant subspaces (and subsets) are ruled out. Even though we do not know of a method to generically rule out weak subspace trails, we do not know of such properties for such a key schedule with random round constants either.

4.4.2. Key-Schedule based on Permutation of the Byte Positions

Another possible category of key schedule exploits permutation of the byte positions: each subkey is the result of a particular permutation applied to the whitening key. A concrete example of key schedule based on permutation has been proposed at 'ToSC/FSE 2018 [KLPS17]. This new key schedule is basically a permutation on the key state byte positions, where the key state update function is defined as follows

$$\begin{pmatrix} 0 & 4 & 8 & 12 \\ 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \end{pmatrix} \rightarrow \begin{pmatrix} 11 & 15 & 3 & 7 \\ 12 & 0 & 4 & 8 \\ 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \end{pmatrix}$$

Regarding security, even though no S-Box nor round constant is used in this key schedule, authors prove more active S-Boxes in the related-key model than for AES-128. However, consider the previous subspace $\mathcal{IS}$ defined in (4.7) and assume that the whitening key belongs to such subspace. It follows that any subkey generated by the previous permutation belongs to such subspace (due to particular symmetries of the permutation), which implies the possibility to set up an "infinitely-long" weak invariant subspace of the form (4.8) for a set of weak-keys.

However, a simple way to avoid such invariant subpsace attack would be to add random round-constants. For completeness, we mention that authors of [KLPS17] also propose to "tweak this design (without increasing the tracking effort) by adding an S-Box layer every round to the entire first row of the key state". Due to the analysis just proposed, this operation does not improve the security against the presented invariant subspace attack. However, this problem can be easily fixed by applying an S-Box layer every round to one entire column.

4.4.3. AES Key-Schedule

Weak-Keys of AES-128. Under one of the 2^{32} weak-keys in K_{weak}

$$K_{\text{weak}} = \left\{ \begin{bmatrix} A & A & A & A \\ B & B & B & B \\ C & C & C & C \\ D & D & D & D \end{bmatrix} \middle| \forall A, B, C, D \in \mathbb{F}_{2^8} \right\} \tag{4.9}$$

the subspace $\mathcal{IS}$ is mapped into a coset of $\mathcal{IS}$ after two complete AES rounds. In more details, given $k \in K_{\text{weak}}$, let $\hat{k}$ be the corresponding subkey after 2 rounds of the key schedule (where $\hat{k} \notin K_{\text{weak}}$ in general). It follows that

$$\mathcal{IS} \xrightarrow{R^2 \circ ARK(\cdot)} \mathcal{IS} \oplus \hat{k}$$

where $R(\cdot) \equiv ARK \circ MC \circ SR \circ$ S-Box$(\cdot)$, that is $\mathcal{IS}$ *forms a weak invariant subspace of length 2.* In order to prove this result, it is sufficient to note that

1. $K_{\text{weak}} \subseteq \mathcal{IS}$, which implies that $\mathcal{IS} \oplus k = \mathcal{IS}$ for all $k \in K_{\text{weak}}$;
2. the first round key derived from the key-schedule of K_{weak} – denoted by K'_w – is a subset of $\mathcal{IS}$

$$K'_w \equiv \begin{bmatrix} \text{S-Box}(B) \oplus A \oplus R[1] & \text{S-Box}(B) \oplus R[1] & \text{S-Box}(B) \oplus A \oplus R[1] & \text{S-Box}(B) \oplus R[1] \\ \text{S-Box}(C) \oplus B & \text{S-Box}(C) & \text{S-Box}(C) \oplus B & \text{S-Box}(C) \\ \text{S-Box}(D) \oplus C & \text{S-Box}(D) & \text{S-Box}(D) \oplus C & \text{S-Box}(D) \\ \text{S-Box}(A) \oplus D & \text{S-Box}(A) & \text{S-Box}(A) \oplus D & \text{S-Box}(A) \end{bmatrix}$$

for all $A, ..., D \in \mathbb{F}_{2^8}$.

Weak-Keys of AES-256. For the case AES-256, a set of 2^{128} weak-keys is given by

$$K_{\text{weak}} = \left\{ \begin{bmatrix} A^0 & A^1 & A^0 & A^1 & E^0 & E^1 & E^0 & E^1 \\ B^0 & B^1 & B^0 & B^1 & F^0 & F^1 & F^0 & F^1 \\ C^0 & C^1 & C^0 & C^1 & G^0 & G^1 & G^0 & G^1 \\ D^0 & D^1 & D^0 & D^1 & H^0 & H^1 & H^0 & H^1 \end{bmatrix} \middle| \forall A^i, \dots, H^i \in \mathbb{F}_{2^8} \, \forall i = 0, 1 \right\}$$

Under any of such keys, the subspace $\mathcal{IS}$ is mapped after two complete rounds into a coset of $\mathcal{IS}$, that is $\mathcal{IS} \xrightarrow{R^2 \circ ARK(\cdot)} \mathcal{IS} \oplus \hat{k}$, where $\hat{k}$ is the corresponding subkey after 2 rounds of the key schedule.

For the follow-up, we also present two subspaces of K_{weak} for which it is possible to construct a longer invariant subspace trail:

3-round: working with any of the 2^{96} keys that satisfy $A^0 = A^1, \dots, D^0 = D^1$, the subspace $\mathcal{IS}$ is mapped after three complete rounds into a coset of $\mathcal{IS}$, that is $\mathcal{IS} \xrightarrow{R^3 \circ ARK(\cdot)} \mathcal{IS} \oplus \hat{k}'$ where $\hat{k}'$ is the subkey after 3 rounds.

4-round: working with any of the 2^{64} keys that satisfy $A^0 = A^1, \dots, H^0 = H^1$, the subspace $\mathcal{IS}$ is mapped after four complete rounds into a coset of $\mathcal{IS}$, that is $\mathcal{IS} \xrightarrow{R^4 \circ ARK(\cdot)} \mathcal{IS} \oplus \hat{k}''$ where $\hat{k}''$ is the subkey after 4 rounds.

5-round: working with any of the 2^{32} keys that satisfy $A^0 = A^1 = B^0 = \ldots = D^0 = D^1 = 0, \quad E^0 = E^1, \dots, H^0 = H^1$, the subspace $\mathcal{IS}$ is mapped after five complete rounds into a coset of $\mathcal{IS}$, that is $\mathcal{IS} \xrightarrow{R^5 \circ ARK(\cdot)} \mathcal{IS} \oplus \hat{k}'''$ where $\hat{k}'''$ is the subkey after 5 rounds.

A complete proof of these results can be found in [GLR+18, App. B].

Weak-Keys of AES-192. For the case AES-192, a set[12] of 2^{64} weak-keys is given by

$$K_{\text{weak}} \equiv \begin{bmatrix} A & E \oplus \text{SB}(D \oplus H) & A & E \oplus \text{SB}(D \oplus H) & E & \text{SB}(D \oplus H) \\ B & F \oplus \text{SB}(A \oplus E \oplus R[1]) & B & F \oplus \widehat{\text{SB}}(A \oplus E) & F & \text{SB}(A \oplus E \oplus R[1]) \\ C & G \oplus \text{SB}(B \oplus F) & C & G \oplus \text{SB}(B \oplus F) & G & \text{SB}(B \oplus F) \\ D & H \oplus \text{SB}(C \oplus G) & D & H \oplus \text{SB}(C \oplus G) & H & \text{SB}(C \oplus G) \end{bmatrix}$$

where $SB(\cdot) \equiv \text{S-Box}^{-1}(\cdot)$ and for each $A, ..., H \in \mathbb{F}_{2^8}$.

Under any of such keys, the subspace $\mathcal{IS}$ is mapped after two complete rounds into a coset of $\mathcal{IS}$, that is $\mathcal{IS} \xrightarrow{R^2 \circ ARK(\cdot)} \mathcal{IS} \oplus \hat{k}$, where $\hat{k}$ the corresponding subkey after 2 rounds of the key schedule.

[12] We highlight that this subset is *not* a subspace, as for AES-128 and AES-256.

4.4.4. Weak-key subspace trail of AES

Before going on, we present a (proper) weak-key subspace trail for AES. The trails that we just proposed in this section are actually invariant subspace trails: here we present *subspace trails with different input and output subspaces that work only for a class of weak keys*, that is weak-key subspace trails which can not be reduced to invariant subspace trails.

For simplicity, initially we work with a simpler S-Box, that is we replace the AES S-Box with the following one

$$\forall x \in GF(2^8): \qquad \text{Sbox}(x) = \begin{cases} 1/x \equiv x^{254}, & \text{if } x \neq 0, \\ 0 & \text{otherwise} \end{cases}$$

To achieve our goal, the idea is to find subspace $V, W \subset GF(2^8)$ of dimension two or/and four such that

$$\text{Sbox}(V \oplus v) \subseteq W \oplus w$$

for *certain* (not all) $v, w \in GF(2^8)$, where $V \neq W$ in general. E.g. the subspace V of dimension four defined as

$$V = \{x \in GF(2^8) \,|\, x^{256} + x = 0\}.$$

It's simple to observe that V is invariant under the Sbox - that is, $Sbox(V) = V$, since $Sbox(x)^{256} \oplus Sbox(x) = [(x^{254})]^{256} \oplus x^{254} = \underbrace{[(x^{254})]^{255}}_{\equiv 1} \cdot x^{254} \oplus x^{254} = 0$ (remember that $x^{2^n-1} = 1$ for all $x \in GF(2^n)$).

In [BWP05], several subspaces $V, W \subset GF(2^8)$ of dimension two and four are defined such that $V \neq W$ and $\text{Sbox}(V \oplus v) \subseteq W \oplus w$. In particular, they found 85 disjoint input subspaces of dimension 2 together with the corresponding output subspaces, and 17 disjoint input subspaces of dimension 4 together with the corresponding output subspaces of the AES[13], e.g.

$$\text{Sbox}\big(V \equiv \langle [2, 24, 97, 160], 0\rangle\big) = \big(W \equiv \langle [6, 40, 88, 139], 0\rangle\big).$$

This can be used to set up a weak-subspace trail for 1-round AES, e.g.

$$\mathcal{V} \oplus x \equiv \begin{bmatrix} \langle [2,24,97,160]\rangle & x_{0,1} & x_{0,2} & x_{0,3} \\ x_{1,0} & \langle [2,24,97,160]\rangle & x_{1,2} & x_{1,3} \\ x_{2,0} & x_{2,1} & \langle [2,24,97,160]\rangle & x_{2,3} \\ x_{3,0} & x_{3,1} & x_{3,2} & \langle [2,24,97,160]\rangle \end{bmatrix} \xrightarrow{MC \circ SR \circ \text{Sbox}(K^w \oplus \cdot)}$$

$$\mathcal{W} \oplus y \equiv \begin{bmatrix} \langle [6,40,88,139]\rangle & y_{0,1} & y_{0,2} & y_{0,3} \\ \langle [6,40,88,139]\rangle & y_{1,1} & y_{1,2} & y_{1,3} \\ \langle [6,40,88,139]\rangle & y_{2,1} & y_{2,2} & y_{2,3} \\ \langle [6,40,88,139]\rangle & y_{3,1} & y_{3,2} & y_{3,3} \end{bmatrix}$$

for random value of $x \in \mathcal{D}_{1,2,3}$, and where the class of weak keys K^w corresponds to the subspace $\mathcal{V} \oplus \mathcal{D}_{1,2,3}$ (where $\mathcal{V} \subseteq \mathcal{D}_0$), that is each byte in the first diagonal of the key belongs to the subspace $\langle [2, 24, 97, 160]\rangle$ while all other bytes can take any possible value.

In a similar way, it is possible to set up different and longer weak-key subspace trails for AES. Finally, we mention that analogous result can be obtained for real AES, since the AES S-Box is affine equivalent to $\text{Sbox}(x)$, that is

$$\text{AES-SBox}(x) = \alpha \cdot \text{Sbox}(x) \oplus \beta \equiv \alpha \cdot \frac{1}{x} \oplus \beta$$

where α is a 8×8 binary (invertible) matrix and β is a constant ($\beta = 0\text{x}63$). In other words, the previous weak-key subspace trail holds if the subspace W is replaced by $\alpha \cdot W \oplus \beta$.

[13] About the notation, the flats are denoted by $\langle [a_1, ..., a_d], b\rangle$, where b represents the coset and $a_1, ..., a_d$ the d basis vectors of the subspace. Here the vectors are denoted by their radius-2 notation, i.e. $x = x_1 + 2 \cdot x_2 + ... + 2^{n-1} \cdot x_n \in \mathbb{Z}$ corresponds with the vector $x = (x_1, ..., x_n)$.

4.5. Weak-Key Truncated Differential for round-reduced AES

In the following, we show that it is possible to extend the truncated differential distinguishers proposed in Sect. 4.3 for up to 5 rounds in the case of weak-key. For simplicity, we focus on the case of AES-128 - analogous results hold for AES-192 and AES-256 for the corresponding class of weak-keys K_{weak}.

As we have just seen, for the case AES-128, the subspace $\mathcal{IS}$ is mapped into a coset $\mathcal{IS} \oplus a$ after two rounds if the secret key is a weak-key. In other words, given two plaintexts $x, y \in \mathcal{IS}$, then $R^2(x) \oplus R^2(y) \in \mathcal{IS}$ under a weak-key. By definition of $\mathcal{IS}$ and of $\mathcal{D}_I$, note that[14]

$$Prob\big[z \in \mathcal{D}_I \,\big|\, z \in \mathcal{IS}\big] = \begin{cases} 2^{-32} & I \equiv \{0,2\}, \{1,3\} \\ 0 & \text{otherwise} \end{cases}$$

where we assume that $z \notin \mathcal{D}_L$ for all $L \subseteq \{0,1,2,3\}$ such that $|L| < |I| < 4$. This is the starting point for our results, together with the fact that $Prob[z \in \mathcal{D}_{0,2}] = Prob[z \in \mathcal{D}_{1,3}] = 2^{-64}$ for a generic text z.

Weak-Key Truncated Differential over 4-round AES-128

Since $R^2(\mathcal{D}_I \oplus a) = \mathcal{M}_I \oplus b$ (that is $Prob\big[R^2(x) \oplus R^2(y) \in \mathcal{M}_I \,\big|\, x \oplus y \in \mathcal{D}_I\big] = 1$), it follows that for an AES permutation and for a weak-key

$$Prob\big[R^4(x) \oplus R^4(y) \in \mathcal{M}_I \,\big|\, x, y \in \mathcal{IS},\, k \in K_{\text{weak}}\big] = 2^{-32} \qquad \text{if } I \equiv \{0,2\}, \{1,3\},$$

while for a random permutation Π the probability is equal to 2^{-64}.

A similar result holds for 4-round AES-192 and for up to 7-round AES-256.

Weak-Key Truncated Differential over 5-round AES-128

Since $Prob\big[x \in \mathcal{C}_J \,\big|\, x \in \mathcal{M}_I\big] = (2^8)^{-4|I| + |I| \cdot |J|}$ as we have just seen, it is possible to set up a 5-round truncated differential distinguisher on 5-round AES for a weak-key.

Proposition 6. *Let $I \subseteq \{0,1,2,3\}$ fixed. The following probability holds:*

$$Prob\big[R^5(x) \oplus R^5(y) \in \mathcal{M}_I \,\big|\, x, y \in \mathcal{IS},\, k \in K_{weak}\big] = 2^{-95+16\cdot|I|} + 2^{-128+32\cdot|I|} \tag{4.10}$$

Since for a random permutation $Prob\big[\Pi(x) \oplus \Pi(y) \in \mathcal{M}_I \,\big|\, x, y \in \mathcal{IS},\, k \in K_{\text{weak}}\big] = (2^{-32})^{4-|I|}$, it is possible to distinguish the two cases.

Proof. In order to compute the previous probability, we recall the *law of total probability.* Given a finite (or countably infinite) partition $B_1, \ldots, B_n$ of a sample space events in a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ s.t. *(1st)* $B_i \cap B_j = \emptyset$ for each $i \neq j$ and s.t. *(2nd)* $\bigcup_i B_i$ is the entire sample space, then

$$Prob(A) = \sum_{i=1}^{n} Prob(A|B_i) \cdot Prob(B_i). \tag{4.11}$$

It follows that for a fixed I:

$$\begin{aligned} &Prob\big[R^5(x) \oplus R^5(y) \in \mathcal{M}_I \mid x, y \in \mathcal{IS},\, k \in K_{\text{weak}}\big] = \\ &= \big\{Prob\big[R^5(x) \oplus R^5(y) \in \mathcal{M}_I \,\big|\, R^4(x) \oplus R^4(y) \in \mathcal{M}'\big] \times Prob\big[R^4(x) \oplus R^4(y) \in \mathcal{M}'\big]\big\} + \\ &+ \big\{Prob\big[R^4(x) \oplus R^4(y) \notin \mathcal{M}'\big] \times Prob\big[R^5(x) \oplus R^5(y) \in \mathcal{M}_I \,\big|\, R^4(x) \oplus R^4(y) \notin \mathcal{M}'\big]\big\} = \\ &= 2^{-64+16\cdot|I|} \cdot 2^{-31} + 2^{-32\cdot(4-|I|)} \cdot (1 - 2^{-31}) \simeq 2^{-95+16\cdot|I|} + 2^{-128+32\cdot|I|} \end{aligned}$$

where $\mathcal{M}' = \mathcal{M}_{0,2} \cup \mathcal{M}_{1,3}$. □

[14] Observe that the first and the third diagonals of each texts in IS are equal, as well as the second and the fourth ones.

To have concrete numbers, if $|I| = 2$, then the probability for 5-round AES-128 is equal to $3 \cdot 2^{-64}$, while for a random permutation it is equal to 2^{-64}. If $|I| = 1$, then the probability for 5-round AES-128 is equal to 2^{-79}, while for a random permutation it is equal to 2^{-96}. Finally, if $|I| = 3$, then the probability for 5-round AES-128 is equal to $2^{-32} + 2^{-47}$, while for a random permutation it is equal to 2^{-32}.

A similar result holds for 5-round AES-192 and for up to 8-round AES-256.

4.6. Generalization of Truncated Differential: Moments of a Probabilistic Distribution

To conclude, we propose a generalization of the truncated differential distinguishers on round-reduced AES just proposed. *The central idea is to consider the variance/skewness/kurtosis/... instead of the mean as probabilistic parameter to set up the distinguisher.* Such strategy can be used for any cipher/hash function. Even if for the studied cases it is not competitive to considerer e.g. the variance instead of the mean, in the following we show a case (see Sect. 5 - truncated differential distinguishers on 5-round AES) in which such strategy is instead competitive.

The strategy is the following: *given plaintexts in a chosen coset of a certain subspace $\mathcal{X}$, the idea is to consider the moments of the probabilistic distribution of the number of corresponding pair of ciphertexts that belong to the same coset of another subspace $\mathcal{Y}$.*

4.6.1. Probabilistic Distributions

Given plaintexts in a chosen coset of a certain subspace $\mathcal{X}$, *what is the probabilistic distribution* of the number of corresponding pair of ciphertexts that belong to the same coset of another subspace $\mathcal{Y}$?

Such probabilistic distribution is - well approximated - by a *binomial distribution*. By definition, a binomial distribution with parameters n and p is the discrete probability distribution of the number of successes in a sequence of n independent yes/no experiments, each of which yields success with probability p. In our case, given n pairs of texts, each one of them satisfies or not the above property/requirement with the *same* probability p. Thus, this model is well described by a binomial distribution. We remember that for a random variable Z that follows the binomial distribution, that is $Z \sim \mathcal{B}(n, p)$, the mean μ, the variance σ^2 and the skewness γ are respectively given by

$$\mu = n \cdot p, \qquad \sigma^2 = n \cdot p \cdot (1-p), \qquad \gamma = \frac{1-2p}{\sqrt{n \cdot p \cdot (1-p)}}.$$

For the follow-up, we remember that a good approximation of the binomial distribution is the normal one if the skewness is equal or close to zero (see "De Moivre–Laplace Theorem" for more details). By definition of skewness, the binomial distribution $\mathcal{B}(n, p)$ is well approximated by a normal one if

$$p = 1/2 \qquad \text{and/or} \qquad n \gg p^{-1}.$$

4.6.2. First Results on round-reduced AES

In the following, we show how to apply the previous results on the truncated differential distinguishers on round-reduced AES. For simplicity, we limit ourselves to consider the subspace trails case, that is the case where the results are independent of the value keys.

Given n chosen plaintexts in the same coset of $\mathcal{D}_I$, we have just seen that the probabilistic distribution of the number of corresponding pair of ciphertexts that belong to the same coset of another subspace $\mathcal{M}_J$ after r rounds AES (for $2 \leq r \leq 4$) and of a random permutation Π are well approximated by a binomial distributions. In order to highlight the differences between the two cases, we assume $|I| = 1$ and we assume both $I, J \subseteq \{1, 2, 3, 4\}$ fixed (which implies $|I| + |J| \leq 4$ for each choice of J).

First, note that it is possible to construct N different couples of texts, where $N = \binom{n}{2} = \frac{n\cdot(n-1)}{2} \approx n^2/2$. Using the probabilities

$$Prob\big[R^r(x) \oplus R^r(y) \in \mathcal{M}_J \,|\, x \oplus y \in \mathcal{D}_I\big]$$
$$Prob\big[\Pi(x) \oplus \Pi(y) \in \mathcal{M}_J \,|\, x \oplus y \in \mathcal{D}_I\big]$$

proposed in this section, we can use the previous formula to compute the mean, the variance and the skewness of these two distributions (for up to 4-round AES). The results are listed in Table 4.1.

Table 4.1.: Given n chosen plaintexts in the same coset of $\mathcal{D}_I$ for $|I| = 1$, in this table we list the moments of the probabilistic distribution of the number of corresponding pair of ciphertexts that belong to the same coset of another subspace $\mathcal{M}_J$ after r-round AES (for $2 \leq r \leq 4$) and of a random permutation Π (where $N = \binom{n}{2} = \frac{n\cdot(n-1)}{2} \approx n^2/2$).

	Mean	**Variance**	**Skewness**
2-round AES	N	0	0
3-round AES	$N \cdot 2^{-32+4\|J\|}$	$N \cdot 2^{-32+4\|J\|} \cdot (1 - 2^{-32+4\|J\|})$	$\frac{1-2^{-31+4\|J\|}}{\sqrt{N\cdot 2^{-32+4\|J\|}\cdot(1-2^{-32+4\|J\|})}}$
4-round AES	0	0	0
$\Pi(\cdot)$	$N \cdot 2^{-128+32\|J\|}$	$N \cdot 2^{-128+32\|J\|} \cdot (1 - 2^{-128+32\|J\|})$	$\frac{1-2^{-127+32\|J\|}}{\sqrt{N\cdot 2^{-128+32\|J\|}\cdot(1-2^{-128+32\|J\|})}}$

Due to the difference between the moments of round-reduced AES and of the random permutation proposed in Table 4.1, *it turns out that every truncated differential distinguishers (and key-recovery attacks) based on the mean just presented in this section can be potentially implemented using the variance and/or the skewness.* Similar considerations can be made for all other cases, and also for the weak subspace trails presented before.

4.6.3. Final Remark: the Pairs of Texts are *not* Independent!

Here we point out an important fact that must be considered is the following: *when considering the number of corresponding pair of ciphertexts that belong to the same coset of another subspace $\mathcal{Y}$, such pairs of ciphertexts are not independent.* Indeed by definition of subspace, given two couples (t^1, t^2) and (t^2, t^3), then if e.g. $t^1 \oplus t^2 \in \mathcal{Y}$ and $t^2 \oplus t^3 \in \mathcal{Y}$, it follows that $t^1 \oplus t^3 \in \mathcal{Y}$ with prob. 1.

Here we show that the previous results are still true even if the pairs are not independent, that is we show that *even if the pairs are not independent, the following probability holds*

$$Prob\big[t^1 \oplus t^2 \in \mathcal{X}\big] = (2^{-32})^{4-\dim(\mathcal{Y})}$$

where $\mathcal{Y}$ is a generic subspace with dimension $\dim(\mathcal{Y})$ and where $t^1 \neq t^2$. For simplicity, we denote the previous probability *pr*.

Consider three texts, that is t^1, t^2 and t^3, and the corresponding three couples, that is (t^1, t^2), (t^1, t^3) and (t^2, t^3). Three possible events can happen:

- if $t^1 \oplus t^2 \in \mathcal{Y}$ and $t^1 \oplus t^3 \in \mathcal{Y}$, then $t^2 \oplus t^3 \in \mathcal{M}_J$ with probability 1 (since $\mathcal{Y}$ is a subspace);
- if $t^1 \oplus t^2 \in \mathcal{Y}$ and $t^1 \oplus t^3 \notin \mathcal{Y}$ (or vice-versa), then $t^2 \oplus t^3 \notin \mathcal{Y}$ with probability 1 (since $\mathcal{Y}$ is a subspace);
- if $t^1 \oplus t^2 \notin \mathcal{Y}$ and $t^1 \oplus t^3 \notin \mathcal{Y}$, then both the events $t^2 \oplus t^3 \in \mathcal{Y}$ and $t^2 \oplus t^3 \notin \mathcal{Y}$ are possible; in particular, $t^2 \oplus t^3 \in \mathcal{Y}$ with *approximately* prob. *pr*.

Thus, *what is the probability that a pair of texts (p, q) satisfy $p \oplus q \in \mathcal{Y}$? In the following, we prove that such probability is equal to pr.*

To answer the previous question, first of all, it is important to focus on the previous last event and to theoretically compute a better approximation of this probability. We are going to show that the last probability is well approximated by $pr \cdot (1-pr)^{-1}$. Since $t^1 \oplus t^2 \notin \mathcal{Y}$, it follows that $4 \cdot [4-\dim(\mathcal{Y})]$ bytes of $t^1 \oplus t^2$ are different from zero, i.e. they can take only $(2^8)^{4\cdot[4-\dim(\mathcal{Y})]} - 1 = pr^{-1} - 1$ possible values different from zero. Similar consideration holds for $t^1 \oplus t^3 \notin \mathcal{Y}$. Since $t^2 \oplus t^3 = (t^1 \oplus t^2) \oplus (t^1 \oplus t^3)$, it follows that the difference on - specific - $4 \cdot [4 - \dim(\mathcal{Y})]$ bytes of $t^2 \oplus t^3$ is equal to zero if the difference on - specific - $4 \cdot [4 - \dim(\mathcal{Y})]$ bytes of $t^1 \oplus t^2$ is equal to the difference on - specific - $4 \cdot [4 - \dim(\mathcal{Y})]$ bytes of $t^1 \oplus t^3$. Since this happens with probability $(pr^{-1} - 1)^{-1}$, it follows that the probability that $t^1 \oplus t^3 \in \mathcal{Y}$ is

$$(pr^{-1} - 1)^{-1} = pr \cdot (1 - pr)^{-1} \approx pr + pr^2 - pr^3 + ...$$

To have more confidence about this fact, note that:

- $t^1 \oplus t^2 \in \mathcal{Y}$, $t^1 \oplus t^3 \in \mathcal{Y}$ and $t^2 \oplus t^3 \in \mathcal{Y}$ occurs with probability pr^2;
- $t^1 \oplus t^2 \in \mathcal{Y}$, $t^1 \oplus t^3 \notin \mathcal{Y}$ and $t^2 \oplus t^3 \notin \mathcal{Y}$ occurs with probability $pr \cdot (1 - pr)$ (similar for the other 3 cases);
- $t^1 \oplus t^2 \notin \mathcal{Y}$, $t^1 \oplus t^3 \notin \mathcal{Y}$ and $t^2 \oplus t^3 \notin \mathcal{Y}$ occurs with probability $(1 - pr)^2 \cdot (1 - pr \cdot (1 - pr)^{-1})$.

All the other cases have probability 0 (since $\mathcal{Y}$ is a subspace). By simple computation, the probability of all the possible events is equal to

$$(pr)^2 + 3 \cdot pr \cdot (1 - pr) + (1 - pr)^2 \cdot (1 - pr \cdot (1 - pr)^{-1}) = 1,$$

as expected. In other words, if one uses the probability $(1 - pr)^3$ for the last case, it follows that the overall probability is less than 1, which is obviously wrong.

Thus, *what is the probability that $t^2 \oplus t^3 \in \mathcal{Y}$?* Using the law of total probability (4.11), it follows that

$$\begin{aligned} Prob\big[t^2 \oplus t^3 \in \mathcal{Y}\big] = & \underbrace{pr \cdot pr \cdot 1}_{1st \text{ Case}} + \underbrace{2 \cdot pr \cdot (1 - pr) \cdot 0}_{2nd \text{ Case}} + \\ & + \underbrace{(1 - pr)^2 \cdot pr \cdot (1 - pr)^{-1}}_{3rd \text{ Case}} = pr. \end{aligned}$$

Since this procedure works for any text t^1, it follows that even if the pairs are not independent, the probability $Prob(t^2 \oplus t^3 \in \mathcal{Y})$ is equal to pr as expected.

5

5-round AES: Probabilistic Distribution

Consider a diagonal set of plaintexts $\mathcal{D}_I \oplus a$, i.e. a set of plaintexts with $|I|$ active diagonal(s). What is the probabilistic distribution of the corresponding number of pairs of ciphertexts after r-round AES that belong to the same coset of $\mathcal{M}_J$ (equivalently, that are equal in $|J|$ anti-diagonal(s), assuming the last MixColumns operation is omitted)?

While a lot is known about the properties of a diagonal set of plaintexts for up to 4-round AES (see Sect. 3 for details), *a complete analysis for 5 or more rounds AES is still missing.* E.g. given a diagonal set of plaintexts and the corresponding ciphertexts after 4 rounds, it is well known that the XOR-sum of the ciphertexts is equal to zero - see integral cryptanalysis [DKR97; KW02], or that each pair of ciphertexts can not be equal in any of the four anti-diagonal (as showed by Biham and Keller in [BK01]).

For the first time, we performed and proposed a precise theoretical differential analysis of such distribution after 5-round AES, supported by practical implementations and verification. In the following, we present in details our results, which are summarized in the following Table.

Table 5.1.: *(Theoretical) Properties of a diagonal set after 5-round encryption.* Given a set of 2^{32} chosen plaintexts all equal in three diagonals (that is, a diagonal set), we consider the *distribution* of the number of different pairs of ciphertexts that lie in a particular subspace $\mathcal{M}_I$ for $I \subseteq \{0,1,2,3\}$ fixed with $|I| = 3$. *Accurate theoretical expected values mean and variance* of this distribution is given in this table for 5-round AES and for a random permutation.

	Random Permutation	**5-round AES**
Mean [GR18]	$2\,147\,483\,647.5 \approx 2^{31}$	$2\,147\,484\,685.6 \approx 2^{31} + 2^{10}$
Variance [GR18]	$2\,147\,483\,647 \approx 2^{31}$	$76\,842\,293\,834.905 \simeq 2^{36.161}$
Multiple-of-8 [GRR17]		✓

Before going on, let us recall some notations that we are going to use often in the following.

Definition 12. *Given two different texts $t^1, t^2 \in \mathbb{F}_{2^b}^{4\times 4}$, we say that $t^1 \leq t^2$ if $t^1 = t^2$ or if there exists $i, j \in \{0,1,2,3\}$ s.t. (1st) $t^1_{k,l} = t^2_{k,l}$ for all $k, l \in \{0,1,2,3\}$ with $k + 4 \cdot l < i + 4 \cdot j$ and (2nd) $t^1_{i,j} < t^2_{i,j}$. Moreover, we say that $t^1 < t^2$ if $t^1 \leq t^2$ (w.r.t. the previous definition) and $t^1 \neq t^2$.*

Definition 13. *Let $\mathcal{X}$ be one of the subspaces previously defined, that is $\mathcal{C}_I$, $\mathcal{D}_I$, $\mathcal{ID}_I$ or $\mathcal{M}_I$. Let $x_0, ..., x_{n-1} \in \mathbb{F}_{2^8}^{4\times 4}$ be a basis of $\mathcal{X}$ - i.e. $\mathcal{X} \equiv \langle x_0, x_1, ..., x_{n-1}\rangle$ where $n = 4 \cdot |I|$ - s.t. $x_i < x_{i+1}$ for each $i = 0, ..., n-1$. Let t be an element of an arbitrary coset of $\mathcal{X}$, that is $t \in \mathcal{X} \oplus a$ for arbitrary $a \in \mathcal{Y}$ (where $\mathcal{Y}$ is the orthogonal subspace of $\mathcal{X}$). We say that t is "generated" by the generating variables $(t^0, ..., t^{n-1})$ - in the following, $t \equiv (t^0, ..., t^{n-1})$ - if and only if*

$$t \equiv (t^0, ..., t^n) \qquad \textit{iff} \qquad t = a \oplus \bigoplus_{i=0}^{n-1} t^i \cdot x_i.$$

As an example, let $\mathcal{X} = \mathcal{M}_0 \equiv \langle MC(e_{0,0}), MC(e_{3,1}), MC(e_{2,2}), MC(e_{1,3})\rangle$, and let $p \in \mathcal{M}_0 \oplus a$. Then $p \equiv (p^0, p^1, p^2, p^3)$ if and only if

$$p \equiv p^0 \cdot MC(e_{0,0}) \oplus p^1 \cdot MC(e_{1,3}) \oplus p^2 \cdot MC(e_{2,2}) \oplus p^3 \cdot MC(e_{3,1}) \oplus a. \tag{5.1}$$

Similarly, let $\mathcal{X} = \mathcal{C}_0 \equiv \langle e_{0,0}, e_{1,0}, e_{2,0}, e_{3,0}\rangle$, and let $p \in \mathcal{C}_0 \oplus a$. Then $p \equiv (p^0, p^1, p^2, p^3)$ if and only if $p \equiv a \oplus p^0 \cdot e_{0,0} \oplus p^1 \cdot e_{1,0} \oplus p^2 \cdot e_{2,0} \oplus p^3 \cdot e_{3,0}$.

5.1. "Multiple-of-8" Property

As first result, we present a new structural property for up to 5 rounds of AES, differential in nature and which is independent of the secret key, of the details of the MixColumns matrix and of the SubBytes operation: *By appropriate choices of difference for a number of input pairs it is possible to make sure that the number of times that the difference of the resulting output pairs lie in a particular subspace is always a multiple of 8.* This "multiple-of-8" property [GRR17] – proposed at Eurocrypt 2017 – allows to set up the first secret-key distinguisher in the literature for 5-round AES which is independent of the secret-key.

Theorem 4 ([GRR17]). *Let $\mathcal{D}_I$ and $\mathcal{M}_J$ the subspaces defined as before for certain fixed I and J, where $1 \le |I| \le 3$. Given an arbitrary coset of $\mathcal{D}_I$ - that is $\mathcal{D}_I \oplus a$ for a fixed $a \in \mathcal{D}_I^\perp$, consider all the $2^{32\cdot|I|}$ plaintexts and the corresponding ciphertexts after 5 rounds, that is (p^i, c^i) for $i = 0, ..., 2^{32\cdot|I|}-1$ where $p^i \in \mathcal{D}_I \oplus a$ and $c^i = R^5(p^i)$. The number n of different pairs of ciphertexts (c^i, c^j) for $i \neq j$ such that $c^i \oplus c^j \in \mathcal{M}_J$ (i.e. c^i and c^j belong to the same coset of $\mathcal{M}_J$)*

$$n := |\{(p^i, c^i), (p^j, c^j) \,|\, \forall p^i, p^j \in \mathcal{D}_I \oplus a,\ p^i < p^j \text{ and } c^i \oplus c^j \in \mathcal{M}_J\}| \tag{5.2}$$

is a multiple of 8, that is $\exists\, n' \in \mathbb{N}$ such that $n = 8 \cdot n'$, independently of the secret-key, of the details of the S-Box and of the MDS MixColumns matrix (except for the branch number equal to 5).

Only for completeness, if the final MixColumns operation is omitted, then the above theorem holds in the same way with $\mathcal{ID}_J$ instead of $\mathcal{M}_J$. Before going on, we also mention that a new framework for proving and adapting the result just proposed has been proposed in [BCC19]. In there, authors re-formulate the above property as immediate consequence of an equivalence relation on the input pairs, under which the difference at the output of the round function is invariant. This approach provides a further understanding of this newly developed distinguisher.

5.1.1. Proof

As we have seen, a coset of $\mathcal{D}_I$ is always mapped into a coset of $\mathcal{C}_I$ after one round and in a coset of $\mathcal{M}_I$ after two rounds, that is for each $a \in \mathcal{D}_I^\perp$ there exists unique $b \in \mathcal{C}_I^\perp$ and unique $c \in \mathcal{M}_I^\perp$ such that $R^2(\mathcal{D}_I \oplus a) = R(\mathcal{C}_I \oplus b) = \mathcal{M}_I \oplus c$. This statement holds also in the same way in the reverse direction, that is for each $b' \in \mathcal{M}_I^\perp$ there exists unique $a' \in \mathcal{D}_I^\perp$ such that $R^{-2}(\mathcal{M}_I \oplus b') = \mathcal{D}_I \oplus a'$. Since

$$\mathcal{D}_I \oplus a \xrightarrow[\text{prob. 1}]{R(\cdot)} \mathcal{C}_I \oplus b \xrightarrow{R^2(\cdot)} \mathcal{D}_J \oplus a' \xrightarrow[\text{prob. 1}]{R^2(\cdot)} \mathcal{M}_J \oplus b',$$

the idea is to focus only on the two central rounds $\mathcal{C}_I \oplus b \to \mathcal{D}_J \oplus a'$ in order to prove the statement of Theorem 4. In particular, this theorem on 5 rounds of AES (and its proof) is related to the following lemma on 2-round AES.

Lemma 4. *Let $\mathcal{C}_I$ and $\mathcal{D}_J$ the subspaces defined as before for certain fixed I and J, where $1 \le |I| \le 3$. Given an arbitrary coset of $\mathcal{C}_I$, consider all the 2^{32} plaintexts and the corresponding ciphertexts after 2 round, that is $(\hat{p}^i, \hat{c}^i)$ for $i = 0, ..., 2^{32\cdot|I|} - 1$ where $\hat{c}^i = R^2(\hat{p}^i)$. The number n of different pairs of ciphertexts $(\hat{c}^i, \hat{c}^j)$ for $i \neq j$ such that $\hat{c}^i \oplus \hat{c}^j \in \mathcal{D}_J$ (i.e. $\hat{c}^i$ and $\hat{c}^j$ belong to the same coset of $\mathcal{D}_J$) is a multiple of 8, that is $\exists\, n' \in \mathbb{N}$ s.t. $n = 8 \cdot n'$.*

We emphasize that the proof of Theorem 4 follows immediately by the proof of Lemma 4. Indeed, note that considering 2^{32} plaintexts in the same coset of $\mathcal{D}_I$ is equivalent to consider 2^{32} texts in the same coset of $\mathcal{C}_I$ after one round. Moreover, note that the number of collisions (i.e. a pair of texts that belong to the same coset of a given subspace) in the same coset of $\mathcal{M}_J$ is equal to the number of collisions in the same coset of $\mathcal{D}_J$ two rounds before.

To prove the lemma, the idea is show that if one pair of ciphertexts satisfies the requirement to belong to the same coset of $\mathcal{D}_J$, then also other pairs of ciphertexts have the same property with probability 1. We highlight that the statement given in Theorem 4 (or Lemma 4) does not depend on the details of the MixColumns matrix or/and of the SubBytes operation.

Proof. For simplicity, we limit ourselves to give all the details for the case $|I| = 1$. The proof for the other cases is analogous – more details are given in the following and in [GRR17].

To prove the desired result[1], we use the "super-Sbox" notation (3.1)

$$\textit{super-Sbox}(\cdot) = \text{S-Box} \circ ARK \circ MC \circ \text{S-Box}(\cdot).$$

For the follow-up, note that the super-Sbox works independently on each column of the texts. As it is well known, 2-round AES can be rewritten as

$$R^2(\cdot) = ARK \circ MC \circ SR \circ \textit{super-Sbox} \circ SR(\cdot).$$

Since ShiftRows and MixColumns operations are linear, it is sufficient to prove the following equivalent result. Given an arbitrary coset of $SR(\mathcal{C}_I)$, consider all the 2^{32} plaintexts in a coset of $SR(\mathcal{C}_I)$ and the corresponding ciphertexts after the super-Sbox, that is $(\tilde{p}^i, \tilde{c}^i = \textit{super-Sbox}(\tilde{p}^i))$ for $i = 0, ..., 2^{32\cdot|I|} - 1$. The number n of different pairs of ciphertexts $(\tilde{c}^i, \tilde{c}^j)$ for $i \neq j$ such that $\tilde{c}^i \oplus \tilde{c}^j \in \mathcal{W}_J$ where

$$\mathcal{W}_J := SR^{-1} \circ MC^{-1}(\mathcal{D}_J). \tag{5.3}$$

is a multiple of 8, that is $\exists\, n' \in \mathbb{N}$ s.t. $n = 8 \cdot n'$.

Consider two elements $\tilde{p}^1$ and $\tilde{p}^2$ in the same coset of $SR(\mathcal{C}_i) \oplus a \equiv \mathcal{ID}_i \oplus a$ for $a \in \mathcal{ID}_i^{\perp}$. Without loss of generality (W.l.o.g.), assume $i = 0$ (it is analogous for the other cases). By definition of $\mathcal{ID}_i$, there exist $x^j, y^j, z^j, w^j \in \mathbb{F}_{2^8}$ for $j = 1, 2$ such that:

$$\tilde{p}^1 = a \oplus \begin{bmatrix} x^1 & 0 & 0 & 0 \\ 0 & 0 & 0 & y^1 \\ 0 & 0 & z^1 & 0 \\ 0 & w^1 & 0 & 0 \end{bmatrix}, \qquad \tilde{p}^2 = a \oplus \begin{bmatrix} x^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & y^2 \\ 0 & 0 & z^2 & 0 \\ 0 & w^2 & 0 & 0 \end{bmatrix}.$$

According to Def. 13, we say that $\tilde{p}^j$ is "generated" by the variables (x^j, y^j, z^j, w^j), that is $\tilde{p}^j \equiv (x^j, y^j, z^j, w^j)$.

Three Equal Generating Variables. Firstly, we consider the case in which three generating variables are equal, e.g. $x^1 \neq x^2$, $y^1 = y^2$, $z^1 = z^2$ and $w^1 = w^2$. Equivalently, this means that we are considering two texts $\tilde{p}^1$ and $\tilde{p}^2$ in the same coset of $\mathcal{ID}_0 \cap \mathcal{C}_0 \subseteq \mathcal{C}_0$, or equivalently $SR^{-1}(\tilde{p}^1)$ and $SR^{-1}(\tilde{p}^2)$ are in the same coset of $\mathcal{D}_0$.

Due to the "impossible differential trail" given in (4.6), we know that

$$\forall J \subseteq \{0,1,2,3\}: \qquad Prob\big[R^2(p^1) \oplus R^2(p^2) \in \mathcal{D}_J \,|\, p^1 \oplus p^2 \in \mathcal{D}_0\big] = 0.$$

As a result, for the case in which three generating variables are equal, then the number of collisions is equal to 0 with prob. 1.

[1] The proof proposed here is not equal to the original one presented in [GRR17].

Two Equal Generating Variables. Secondly, we consider the case in which two generating variables are equal, e.g. $x^1 \neq x^2$, $y^1 \neq y^2$, $z^1 = z^2$ and $w^1 = w^2$. Equivalently, this means that we are considering two texts $\hat{p}^1$ and $\hat{p}^2$ in the same coset of $\mathcal{ID}_0 \cap \mathcal{C}_{0,1} \subseteq \mathcal{C}_{0,1}$, or equivalently $SR^{-1}(\tilde{p}^1)$ and $SR^{-1}(\tilde{p}^2)$ are in the same coset of $\mathcal{D}_{0,1}$.

Due to the "impossible differential trail" given in (4.6), the event $R^2(p^1) \oplus R^2(p^2) \in \mathcal{D}_J$ is possible if and only if $|J| \geq 3$. In other words, if $|J| \leq 2$ and for the case in which three generating variables are equal, then the number of collisions is equal to 0 with prob. 1.

Consider the case $|J| \geq 3$. *Since each column of $\tilde{p}^1$ and $\tilde{p}^2$ depends on different and independent variables, since the super-Sbox works independently on each column and since the XOR-sum is commutative*, it follows that

$$\textit{super-Sbox}(\tilde{p}^1) \oplus \textit{super-Sbox}(\tilde{p}^2) = \textit{super-Sbox}(q^1) \oplus \textit{super-Sbox}(q^2)$$

where q^1, q^2 are texts in the same coset $SR(\mathcal{C}_i) \oplus a$ generated by:

$$1.\,(x^1, y^1, z, w) \text{ and } (x^2, y^2, z, w); \qquad 2.\,(x^2, y^1, z, w) \text{ and } (x^1, y^2, z, w);$$

for each $z, w \in \mathbb{F}_{2^8}$. Indeed, note that if the second and the third columns of $\tilde{p}^1$ and of $\tilde{p}^2$ are equal, then the second and the third columns of *super-Sbox*$(\tilde{p}^1) \oplus$ *super-Sbox*$(\tilde{p}^2)$ are equal to zero independently of the value of z and w. As a result, for the case in which two generating variables are equal, then the number of collisions is a multiple of $2 \cdot (2^8)^2 = 2^{17}$.

One Equal Generating Variable. Thirdly, we consider the case in which one generating variable is equal, e.g. $x^1 \neq x^2$, $y^1 \neq y^2$, $z^1 \neq z^2$ and $w^1 = w^2$. Equivalently, this means that we are considering two texts $\tilde{p}^1$ and $\tilde{p}^2$ in the same coset of $\mathcal{ID}_0 \cap \mathcal{C}_{0,1,2} \subseteq \mathcal{C}_{0,1,2}$, or equivalently $SR^{-1}(\tilde{p}^1)$ and $SR^{-1}(\tilde{p}^2)$ are in the same coset of $\mathcal{D}_{0,1,2}$.

Due to the "impossible differential trail" given in (4.6), the event $R^2(p^1) \oplus R^2(p^2) \in \mathcal{D}_J$ is possible if and only if $|J| \geq 2$. In other words, if $|J| \leq 1$ and for the case in which three generating variables are equal, then the number of collisions is equal to 0 with prob. 1.

Consider the case $|J| \geq 2$. *Since each column of $\tilde{p}^1$ and $\tilde{p}^2$ depends on different and independent variables, since the super-Sbox works independently on each column and since the XOR-sum is commutative*, it follows that

$$\textit{super-Sbox}(\tilde{p}^1) \oplus \textit{super-Sbox}(\tilde{p}^2) = \textit{super-Sbox}(q^1) \oplus \textit{super-Sbox}(q^2)$$

where q^1, q^2 are texts in the same coset $SR(\mathcal{C}_i) \oplus a$ generated by:

$$\begin{array}{ll} 1.\,(x^1, y^1, z^1, w) \text{ and } (x^2, y^2, z^2, w); & 2.\,(x^2, y^1, z^1, w) \text{ and } (x^1, y^2, z^2, w); \\ 3.\,(x^1, y^2, z^1, w) \text{ and } (x^2, y^1, z^2, w); & 4.\,(x^1, y^1, z^2, w) \text{ and } (x^2, y^2, z^1, w); \end{array}$$

for each $w \in \mathbb{F}_{2^8}$. Indeed, note that if the second column of $\tilde{p}^1$ and of $\tilde{p}^2$ are equal, then the second column of *super-Sbox*$(\tilde{p}^1) \oplus$ *super-Sbox*$(\tilde{p}^2)$ are equal to zero independently of the value of w. As a result, for the case in which one generating variable is equal, then the number of collisions is a multiple of $4 \cdot 2^8 = 2^{10}$.

Different Generating Variables. Finally we consider the case in which all generating variables are different, that is $x^1 \neq x^2$, $y^1 \neq y^2$, $z^1 \neq z^2$ and $w^1 \neq w^2$.

Since each column of $\tilde{p}^1$ and $\tilde{p}^2$ depends on different and independent variables, since the super-Sbox works independently on each column and since the XOR-sum is commutative, it follows that

$$\textit{super-Sbox}(\tilde{p}^1) \oplus \textit{super-Sbox}(\tilde{p}^2) = \textit{super-Sbox}(q^1) \oplus \textit{super-Sbox}(q^2)$$

where q^1, q^2 are texts in the same coset $SR(\mathcal{C}_i) \oplus a$ generated by:

1. (x^1, y^1, z^1, w^1) and (x^2, y^2, z^2, w^2);
2. (x^2, y^1, z^1, w^1) and (x^1, y^2, z^2, w^2);
3. (x^1, y^2, z^1, w^1) and (x^2, y^1, z^2, w^2);
4. (x^1, y^1, z^2, w^1) and (x^2, y^2, z^1, w^2);
5. (x^1, y^1, z^1, w^2) and (x^2, y^2, z^2, w^1);
6. (x^2, y^2, z^1, w^1) and (x^1, y^1, z^2, w^2);
7. (x^2, y^1, z^2, w^1) and (x^1, y^2, z^1, w^2);
8. (x^2, y^1, z^1, w^2) and (x^1, y^2, z^2, w^1).

As a result, for the case in which all generating variables are different, then the number of collisions is a multiple of 8.

Conclusion. The "multiple-of-8" property follows immediately by combining the results just given.
□

Without going into the details, we discuss the case $|I| \geq 2$. W.l.o.g consider $|I| = 2$ and assume $I = \{0, 1\}$ (the other cases are analogous). The proof works exactly as before.

Given two texts p and q in the same coset of $SR(\mathcal{C}_I)$, that is $SR(\mathcal{C}_I) \oplus a$ for a given $a \in SR(\mathcal{C}_I)^\perp$, there exist $p_0^{'}, p_0^{''}, p_1^{'}, p_1^{''}, p_2^{'}, p_2^{''}, p_3^{'}, p_3^{''} \in \mathbb{F}_{2^8}$ and $q_0^{'}, q_0^{''}, q_1^{'}, q_1^{''}, q_2^{'}, q_2^{''}, q_3^{'}, q_3^{''} \in \mathbb{F}_{2^8}$ such that:

$$p = a \oplus \begin{bmatrix} p_0^{'} & p_1^{''} & 0 & 0 \\ p_0^{''} & 0 & 0 & p_3^{'} \\ 0 & 0 & p_2^{'} & p_3^{''} \\ 0 & p_1^{'} & p_2^{''} & 0 \end{bmatrix}, \qquad q = a \oplus \begin{bmatrix} q_0^{'} & q_1^{''} & 0 & 0 \\ q_0^{''} & 0 & 0 & q_3^{'} \\ 0 & 0 & q_2^{'} & q_3^{''} \\ 0 & q_1^{'} & q_2^{''} & 0 \end{bmatrix}.$$

As for the case $|I| = 1$, the idea is to consider all possible combinations of the variables $p_0 \equiv (p_0^{'}, p_0^{''}), p_1 \equiv (p_1^{'}, p_1^{''}), p_2 \equiv (p_2^{'}, p_2^{''}), p_3 \equiv (p_3^{'}, p_3^{''})$ and $q_0 \equiv (q_0^{'}, q_0^{''}), q_1 \equiv (q_1^{'}, q_1^{''}), q_2 \equiv (q_2^{'}, q_2^{''}), q_3 \equiv (q_3^{'}, q_3^{''})$. In other words, the idea is to consider variables in $(\mathbb{F}_{2^8})^2 \equiv \mathbb{F}_{2^8} \times \mathbb{F}_{2^8}$ and not in $\mathbb{F}_{2^8}$. For $|I| = 3$, the idea is to work with variables in $(\mathbb{F}_{2^8})^3$.

5.1.2. "Multiple-of-8" Secret-Key Distinguisher

Our 5-round distinguisher exploits the property just described that the above defined number n is a multiple of 8 for 5-round AES, while it can take any possible value in the case of a random permutation. In the following we show how to set up the previous distinguisher in an efficient way for the case $|I| = 1$ and $|J| = 3$. In this case, the data cost of the distinguisher is of 2^{32} chosen plaintexts, while the computational cost is well approximated by $2^{35.6}$ table look-ups, or equivalently 2^{29} five-round encryptions of AES (using the approximation 20 table look-ups $\approx$ 1 round of AES). We emphasize that *this is the first and currently the most competitive secret-key distinguisher for 5-round AES in the literature which is independent of the secret key* and that does not require adaptive chosen plaintexts/ciphertexts.

Data Cost. To implement the distinguisher, one has to count the number of pairs of ciphertexts for which the difference in $4 - |J| = 1$ anti-diagonal is equal to zero (where this anti-diagonal is fixed in advance). First of all, since the probability that two ciphertexts satisfy this property is 2^{-32}, we expect that *on average*

$$\binom{2^{32}}{2} \cdot 2^{-32} = 2^{31} \cdot (2^{32} - 1) \cdot 2^{-32} \simeq 2^{31}$$

different pairs of ciphertexts have difference zero in one fixed anti-diagonal both for an AES permutation and for a random one. However, while for an AES permutation this number is a multiple of 8 with probability 1, for a random permutation this happens only with probability $0.125 \equiv 2^{-3}$. In particular, consider s initial arbitrary diagonal sets of plaintexts and for each of them count the number of different pairs of ciphertexts that have difference zero in d anti-diagonals. For an AES permutation, each of these numbers is a multiple of 8, while the probability that this happens for a random permutation is only $2^{-3\cdot s}$. In order to distinguish the AES permutation from the random one with probability at least pr, it is sufficient that for a random permutation at least one of these numbers is not a multiple of 8, which happens with probability $pr = 1 - 2^{-3\cdot s}$. As a result, the probability of success of this distinguisher is greater than 99% (i.e. $pr \geq 0.99$) for $s \geq 3$. Note that for each initial diagonal set, one can count the above defined number n for each one of the four possible anti-diagonals. In other words, there are four different anti-diagonals for which one can count the number n of pairs of ciphertexts with zero difference in that anti-diagonal. It follows that using a single initial diagonal set, it is possible to distinguish 5-round AES from a random permutation with a probability of success of approximately $1 - (2^{-12}) = 99.975\%$.

In conclusion, 2^{32} chosen plaintexts in the same coset of $\mathcal{D}_I$ for $I \subseteq \{0, 1, 2, 3\}$ fixed with $|I| = 1$ are sufficient to distinguish a random permutation from 5-round AES.

Computational Cost. We have just seen that 2^{32} chosen plaintexts in a single diagonal set are sufficient to distinguish a random permutation from 5 rounds of AES, simply counting the number of pairs of ciphertexts with equal bytes in d anti-diagonal and checking if it is a multiple of 8 or not. Here we give an estimation of the computational cost of the distinguisher, which is approximately given by the sum of the cost to construct all the pairs and of the cost to count the number of pairs of ciphertexts with the previous property. As a result, the total computational cost can be well approximated by $2^{35.6}$ table look-ups.

Assume the final MixColumns operation is omitted. As we have just said, for each initial diagonal set the two steps of the distinguisher are (1) construct all the possible pairs of ciphertexts and (2) count the number of collisions. First of all, given pair of ciphertexts, note that the cost to check that the bytes in d anti-diagonals are equal corresponds to the cost of a XOR operation[2]. As we are going to show, the major cost of this distinguisher regards the construction of all the possible different pairs, which corresponds to step (1). Since it is possible to construct approximately 2^{63} pairs for each initial diagonal set, the simplest way to do it requires 2^{63} table look-ups. In the following, we present a way to reduce the total cost to approximately $2^{35.6}$ table look-ups, where the used tables are of size 2^{32} texts (or equivalently $2^{32} \cdot 16 = 2^{36}$ byte).

The basic idea is to implement the distinguisher using a *data structure*. The goal is to count the number of pairs of ciphertexts (c^1, c^2) for which the bytes in one of the anti-diagonal are equal, that is such that for a fixed $j \in \{0, 1, 2, 3\}$ the following condition is satisfied:

$$c^1_{i,j-i} = c^2_{i,j-i} \qquad \forall i = 0, 1, 2, 3 \tag{5.4}$$

where the index is computed modulo 4. To do this, consider an array A of 2^{32} elements completely initialized to zero. The element of A in position x for $0 \leq x \leq 2^{32} - 1$ - denote by $A[x]$ - represents the number of ciphertexts c that satisfy the following equivalence (in the integer field $\mathbb{N}$): $x = c_{0,0-j} + 256 \cdot c_{1,1-j} + c_{2,2-j} \cdot 256^2 + c_{3,3-j} \cdot 256^3$. It's simple to observe that if two ciphertexts c^1 and c^2 satisfy (5.4), then they increment the same element x of the array A. It follows that given $r \geq 0$ texts that increment the same element x of the array A, then it is possible to construct

$$\binom{r}{2} = \frac{r \cdot (r-1)}{2}$$

different pairs of texts that satisfy (5.4). The complete pseudo-code is given in Algorithm 1.

What is the total computational cost of this procedure? Given a set of 2^{32} (plaintexts, ciphertexts) pairs, one has first to fill the array A using the strategy just described, and then to compute the

[2] As example, let $J \subseteq \{0, 1, 2, 3\}$ with $d = |J|$. Given a pair (c^1, c^2), this operation can be reduced to check that $\tilde{c}_{k,j-k} = 0$ for each $k = 0, ..., 3$ and $j \in J$, where $\tilde{c} \equiv c^1 \oplus c^2$.

```
Data: $2^{32}$ (plaintext, ciphertext) pairs $(p^i, c^i)$ for $i = 0, ..., 2^{32} - 1$ in a single diagonal set.
Result: 1 for an AES permutation, 0 otherwise (prob. $\geq 99\%$)
Let $(p^i, c^i)$ for $i = 0, ..., 2^{32} - 1$ the (plaintext, ciphertext) pairs;
for all $j \in \{0, 1, 2, 3\}$ do
    Let $A[0, ..., 2^{32} - 1]$ an array initialized to zero;
    for $i$ from 0 to $2^{32} - 1$ do
        $x \leftarrow \sum_{k=0}^{3} c^i_{k,j-k} \cdot 256^k$;    // $c^i_{k,j-k}$ denotes the byte of the ciphertext$c^i$ in
         row $k$ and column $j - k \mod 4$
        $A[x] \leftarrow A[x] + 1$;  // $A[x]$ denotes the value stored in the $x$-th address of
         the array $A$
    end
    $n \leftarrow \sum_{i=0}^{2^{32}-1} A[i] \cdot (A[i] - 1)/2$;
    if $(n \mod 8) \neq 0$ then
        return 0;
    end
end
return 1.
```

Algorithm 1: *Secret-Key Distinguisher for 5 rounds of AES* which exploits a property which is independent of the secret key - probability of success: $\geq 99\%$.

number of total of pairs of ciphertexts that satisfy the property, for a cost of $3 \cdot 2^{32} = 2^{33.6}$ table look-ups - each one of these three operations require 2^{32} table look-ups. Since one has to repeat this algorithm 4 times - one time for each one of the four anti-diagonal, the total cost is of $4 \cdot 2^{33.6} = 2^{35.6}$ table look-ups, or equivalently 2^{29} five-round encryptions of AES (using the approximation[3] 20 table look-ups $\approx$ 1 round of AES).

Practical Verification. The proposed distinguisher has also been practically verified on real AES[4]. The practical results of our experiments are in accordance with the theoretical ones.

5.2. Probabilistic Distribution for 5-round AES

Several open questions arise from the "multiple-of-8" result provided in [GRR17]. In particular, given a set of $2^{32 \cdot |I|}$ plaintexts in the same coset of $\mathcal{D}_I$, consider the probabilistic distribution of the number of pairs of ciphertexts which are equal in n fixed anti-diagonal(s) (assuming the final MixColumns operation has been omitted) for $1 \leq n \leq 3$:

- is it possible to say something about the mean, the variance and the skewness of this distribution?
- *does the multiple-of-8 property influence e.g. the average number of output pairs that lie in a particular subspace (i.e. the mean)? Are other parameters (e.g. the variance, the skewness, ...) affected by the multiple-of-8 property?*

In the following, we (partially) answer these questions. Using the multiple-of-8 property just recalled and the results that we are going to present, we can formally describe the probabilistic distribution

[3] We highlight that even if this approximation is not formally correct - the size of the table of an S-Box look-up is lower than the size of the table used for our proposed distinguisher, it allows to give a comparison between our proposed distinguisher and the others currently present in literature. At the same time, we note that the same approximation is largely used in literature.

[4] The source codes of the distinguishers/attacks are available at `https://github.com/Krypto-iaik/AES_5round_SKdistinguisher`

of the number of pairs of ciphertexts which are equal in n fixed anti-diagonal(s) (assuming the final MixColumns operation has been omitted) after 5-round AES, whose corresponding plaintexts are in the same coset of $\mathcal{D}_i$.

Theorem 5 ([GR18]). *Consider an AES-like cipher that works with texts in $\mathbb{F}_{2^8}^{4\times 4}$, s.t. the MixColumns matrix is an MDS matrix and such that the solutions of the equation*

$$\textit{S-Box}(x \oplus \Delta_I) \oplus \textit{S-Box}(x) = \Delta_O. \tag{5.5}$$

are uniformly distributed for each input/output difference $\Delta_I \neq 0$ and $\Delta_O \neq 0$.

Consider 2^{32} plaintexts p^i for $i = 0, 1, ..., 2^{32} - 1$ in a coset of a diagonal space $\mathcal{D}_i$, that is $\mathcal{D}_i \oplus a$ for $i \in \{0, 1, 2, 3\}$ and $a \in \mathcal{D}_i^\perp$, and the corresponding ciphertexts after 5 rounds, that is $c^i = R^5(p^i)$. The distribution probability 5-AES of the number of different pairs of ciphertexts (c^i, c^j) with $c^i < c^j$ that belong to the same coset of $\mathcal{M}_J$ for $J \subseteq \{0, 1, 2, 3\}$ fixed with $|J| = 3$ is described by

$$\textit{5-AES} = 2^3 \times X_3 + 2^{10} \times X_{10} + 2^{17} \times X_{17}$$

where

$$\forall i = 3, 10, 17 : \qquad X_i \sim \mathcal{B}(n_i, p_i)$$

are binomial distributions s.t.

$$\begin{aligned} n_3 &= 2^{28} \cdot (2^8 - 1)^4 & \qquad p_3 &= 2^{-32} + 2^{-53.98306}; \\ n_{10} &= 2^{23} \cdot (2^8 - 1)^3 & p_{10} &= 2^{-32} - 2^{-45.98874}; \\ n_{17} &= 3 \cdot 2^{15} \cdot (2^8 - 1)^2 & p_{17} &= 2^{-32} + 2^{-37.98588}. \end{aligned}$$

The distribution probability 5-AES of the number of different pairs of ciphertexts (c^i, c^j) with $c^i < c^j$ that belong to the same coset of $\mathcal{M}_J$ for $J \subseteq \{0, 1, 2, 3\}$ fixed with $|J| = 3$ has mean value $\mu = 2\,147\,484\,685.6$ and standard deviation $\sigma = 277\,204.426$.

If the final MixColumns is omitted, it is sufficient to replace the mixed space $\mathcal{M}_J$ with an inverse-diagonal space $\mathcal{ID}_J$. We remember that a coset of $\mathcal{D}_k$ corresponds to a set of 2^{32} texts with one active diagonal, while that two ciphertexts c^i and c^h belong to the same coset of an inverse-diagonal space $\mathcal{ID}_J = MC^{-1}(\mathcal{M}_J)$ (that is, $c^i \oplus c^h \in \mathcal{ID}_J$) if and only if they are equal in the j-th anti-diagonal where $j \equiv \{0, 1, 2, 3\} \setminus J$. For completeness, the same result holds also in the decryption direction (that is, using chosen ciphertexts instead of chosen plaintexts).

Lemma 5 ([GR18]). *Consider an AES-like cipher that works with texts in $\mathbb{F}_{2^8}^{4\times 4}$ and for which the assumptions of Theorem 5 hold.*

Consider 2^{32} plaintexts p^i for $i = 0, 1, ..., 2^{32} - 1$ in a coset of a diagonal space $\mathcal{D}_i$, that is $\mathcal{D}_i \oplus a$ for $i \in \{0, 1, 2, 3\}$ and $a \in \mathcal{D}_i^\perp$, and the corresponding ciphertexts after 5 rounds, that is $c^i = R^5(p^i)$. The probability to have $n \in \mathbb{N}$ different pairs of ciphertexts (c^i, c^j) with $c^i < c^j$ that belong to the same coset of $\mathcal{M}_J$ for $J \subseteq \{0, 1, 2, 3\}$ fixed with $|J| = 3$ (i.e. n collisions) is given by:

$$\textit{Prob}(n) = \begin{cases} 0 & \textit{if } n \textit{ mod } 8 \neq 0 \\ \sum_{(k_3, k_{10}, k_{17}) \in K_n} \left[\prod_{i \in \{3,10,17\}} \underbrace{\binom{n_i}{k_i} \cdot (p_i)^{k_i} \cdot (1 - p_i)^{n_i - k_i}}_{\sim \mathcal{B}(n_i, p_i)} \right] & \textit{otherwise} \end{cases}$$

where

$$K_n = \left\{ (k_3, k_{10}, k_{17}) \in \mathbb{N} \times \mathbb{N} \times \mathbb{N} \,\middle|\, 0 \leq k_i \leq n_i \;\textit{ and }\; 2^3 \cdot k_3 + 2^{10} \cdot k_{10} + 2^{17} \cdot k_{17} = n \right\}$$

and where n_i and p_i for $i = 3, 10, 17$ are given in Theorem 5.

Note that $\textit{Prob}(n > [2^3 \cdot n_3 + 2^{10} \cdot n_{10} + 2^{17} \cdot n_{17}]) = 0$.

5.2.1. Sketch of the Proof

The results of Theorem 5 and Lemma 5 are due to the following considerations. First of all, given 2^{32} plaintexts in a coset of $\mathcal{D}_i$, *the corresponding pairs of texts are not independent/unrelated.* Due to the multiple-of-8 property [GRR17] and of the mixture differential cryptanalysis [Gra17b; Gra18b], we know that these pairs of texts can be divided in sets of cardinality respectively 8 (if the generating variables of a pair of texts are all different after 1-round encryption), 2^{10} (if one out of the four generating variables is equal for the pair of texts after 1-round encryption), 2^{17} (if two out of the four generating variables are equal for the pair of texts) and 2^{24} (if three out of the four generating variables are equal for the pair of texts after 1-round encryption) such that

1. pairs of texts of different sets are independent;
2. all pairs in the same set belong or not belong to the same coset of $\mathcal{M}$ after 5 rounds.

In other words, given a set of pairs as just defined, it is *not* possible that the ciphertexts of some pairs belong to the same coset of $\mathcal{M}$ after 5 rounds, while the ciphertexts of other pairs not (see Sect. 6 – "Mixture Differential" distinguisher – for more details). Moreover, let us recall that if three out of the four generating variables of the plaintexts are equal after 1-round encryption, then the corresponding ciphertexts cannot belong to the same coset of $\mathcal{M}$. It follows that the probability of the event "$n = 8 \cdot \hat{n}$ collisions" corresponds to the sum of the probabilities to have "$2^3 \cdot k_3$ collisions in the first set *and* $2^{10} \cdot k_{10}$ collisions in the second set *and* $2^{17} \cdot k_{17}$ collisions in the third set" *for each* k_3, k_{10}, k_{17} *such that* $2^3 \cdot k_3 + 2^{10} \cdot k_{10} + 2^{17} \cdot k_{17} = n$.

Each one of these (independent) events is well characterized by a *binomial distribution.* Indeed, remember that a binomial distribution with parameters n and p is the discrete probability distribution of the number of successes in a sequence of n independent yes/no experiments, each of which yields success with probability p. In our case, given n pairs of texts, each one of them satisfies or not the above property/requirement with the same probability p. For a random variable that follows the binomial distribution $\mathcal{B}(n,p)$, the mean μ and the variance σ^2 are respectively given by $\mu = n \cdot p$ and $\sigma^2 = n \cdot p \cdot (1-p)$.

As a result, the distribution 5-AES of the number of collisions for the AES case is given by

$$\text{5-AES} = 2^3 \times X_3 + 2^{10} \times X_{10} + 2^{17} \times X_{17}$$

where $X_i \sim \mathcal{B}(n_i, p_i)$ for $i = 3, 10, 17$ are binomial distributions.

While the values of n_i can be easily derived using the Multiple-of-8 property and/or the Mixture Diff. distinguisher, a formal computation to derive the probabilities p_i for $i = 3, 10, 17$, the value of the mean and the variance, and the probability given in Lemma 5 will be computed in the following sections.

Preliminary Considerations. First of all, given a coset of $\mathcal{C}_i$ of 2^{32} chosen plaintexts, we compute the number of different pairs of texts with v equal generating variables for $0 \leq v \leq 3$. Note that given a coset of $\mathcal{D}_i$ of 2^{32} chosen plaintexts, one can construct $2^{31} \cdot (2^{32} - 1) \simeq 2^{63}$ different pairs. Among them, the number of pairs of texts with $0 \leq v \leq 3$ equal generating variables (and $4 - v$ different generating variables) *after one round* is given by

$$\binom{4}{v} \cdot 2^{31} \cdot (2^8 - 1)^{4-v}. \tag{5.6}$$

Indeed, if v variables are equal for the two texts of the pair, then they can take $(2^8)^v$ different values. For each one of the remaining $4 - v$ variables, the variables must be different for the two texts. Thus, these $4 - v$ variables can take exactly $\left[2^8 \cdot (2^8 - 1)\right]^{4-v}/2$ different values. The result follows immediately since there are $\binom{4}{v}$ different combinations of v variables.

It follows that

- the number of pairs of texts with "no equal generating variables" is given by $\binom{4}{0} \cdot 2^{31} \cdot (2^8 - 1)^4$. Due to the multiple-of-8 property, note that these pairs can be divided in sets of cardinality 2^3 such that: *(1st)* pairs of texts of different sets are independent and *(2nd)* all pairs in the same set belong or not belong to the same coset of $\mathcal{M}$ after 5 rounds. It follows that the number n_3 of sets of cardinality 2^3 for which *(1st)* all generating variables of each pair of texts are different and *(2nd)* the pairs of texts in the same texts share the same generating variables (in different combinations) is given by
$$n_3 = \frac{1}{8} \cdot \binom{4}{0} \cdot 2^{31} \cdot (2^8 - 1)^4 = 2^{28} \cdot (2^8 - 1)^4;$$
- the number of pairs of texts with "one equal (and three different) generating variable(s)" is given by $\binom{4}{1} \cdot 2^{31} \cdot (2^8 - 1)^3$. As before and due to the multiple-of-8 property, the number n_{10} of sets of cardinality 2^{10} for which *(1st)* one generating variable of each pair of texts is equal and – a part from this one – *(2nd)* the pairs of texts in the same texts share the same generating variables (in different combinations)
$$n_{10} = \frac{1}{2^{10}} \cdot \binom{4}{1} \cdot 2^{31} \cdot (2^8 - 1)^3 = 2^{23} \cdot (2^8 - 1)^3$$
- the number of pairs of texts with "two equal (and two different) generating variable" is given by $\binom{4}{2} \cdot 2^{31} \cdot (2^8 - 1)^2$. As before and due to the multiple-of-8 property, the number n_{17} of sets of cardinality 2^{17} for which *(1st)* two generating variables of each pair of texts are equal and – a part from these ones – *(2nd)* the pairs of texts in the same texts share the same generating variables (in different combinations)
$$n_{17} = \frac{1}{2^{17}} \cdot \binom{4}{2} \cdot 2^{31} \cdot (2^8 - 1)^2 = 3 \cdot 2^{15} \cdot (2^8 - 1)^3$$

5.2.2. About the "Uniform Distribution of Solutions of eq. (5.5)"

Before going on, we discuss the assumptions of Theorem 5, focusing on the assumption related to the properties/details of the S-Box.

Preliminary. Since the assumptions of Theorem 5 depends on the details of the S-Box, we first recall some properties of the S-Box function.

Given a bijective S-Box function, let $\Delta_I, \Delta_O \in \mathbb{F}_{2^8}$. We denote by n_{Δ_I,Δ_O} the number of solutions x of the following equation
$$\text{S-Box}(x \oplus \Delta_I) \oplus \text{S-Box}(x) = \Delta_O.$$

In the following, we limit to consider the cases $\Delta_I \neq 0$ and $\Delta_O \neq 0$ - if $\Delta_O = 0$, the equation admits solution if and only if $\Delta_I = 0$ (the S-Box is bijective).

Independently of the details of the S-Box, the *mean value*[5] of n_{Δ_I,Δ_O} is equal to
$$\mathbb{E}[n_{\Delta_I,\Delta_O}] = \frac{256}{255} \simeq 1.00392... \simeq 1 + 2^{-7.9944}, \tag{5.7}$$

Indeed, observe that for each x and for each $\Delta_I \neq 0$ there exists $\Delta_O \neq 0$ (since S-Box is bijective) that satisfies eq. (5.5). Since there are 256 different x and 255 different values of Δ_I and Δ_O, the average number of solutions is $\frac{256 \cdot 255}{255^2} = \frac{256}{255}$ independently of the details of the (bijective) S-Box.

[5] In the case of a discrete probability distribution of a random variable X, the mean $E[X] \equiv \mu$ is defined as $\mu = \sum x \cdot P(x)$, i.e. the sum over every possible value x weighted by the probability of that value $P(x)$.

In the following, we denote by $Var(n_{\Delta_I,\Delta_O})$ the *variance*[6] of n_{Δ_I,Δ_O}. This quantity *depends on the details of the S-Box*, in particular on the distribution of n_{Δ_I,Δ_O} with respect to Δ_I and Δ_O. For the AES S-Box case, for each $\Delta_I \neq 0$ there are 128 values of $\Delta_O \neq 0$ for which equation (5.5) has no solution, 126 values of $\Delta_O \neq 0$ for which equation (5.5) has 2 solutions ($\hat{x}$ is a solution iff $\hat{x} \oplus \Delta_I$ is a solution) and finally 1 value of $\Delta_O \neq 0$ for which equation (5.5) has 4 solutions. The variance of the AES S-Box is so equal to $Var_{AES}(n_{\Delta_I,\Delta_O}) = 2^2 \cdot \frac{126}{255} + 4^2 \cdot \frac{1}{255} - \left(\frac{256}{255}\right)^2 = \frac{67\,064}{65\,025}$.

The *Maximum Differential Probability DP_{max}* of an S-Box is defined as

$$DP_{max} = \max_{\Delta_I \neq 0, \Delta_O} \frac{n_{\Delta_I,\Delta_O}}{2^n}. \tag{5.8}$$

Since all entries of the differential distribution table are even, DP_{max} is always bigger than or equal to 2^{-n+1} (i.e. $DP_{max} \geq 2^{-n+1}$). *Permutations with $DP_{max} = 2^{-n+1}$ are called Almost Perfect Nonlinear (APN).*

Finally, given $\Delta_I \neq 0$ (resp. $\Delta_O \neq 0$), consider the probabilistic distribution of n_{Δ_I,Δ_O} w.r.t. $\Delta_O \neq 0$ (resp. $\Delta_I \neq 0$). We say that the S-Box is "*Uniform*" if such distribution is independent of Δ_I (resp. Δ_O). As examples, the AES S-Box is uniform differential since for each $\Delta_I \neq 0$ (fixed), $Prob(n_{\Delta_I,\Delta_O} = 2) = \frac{126}{255}$ and $Prob(n_{\Delta_I,\Delta_O} = 4) = \frac{1}{255}$. The PRINCE S-Box is instead not uniform differential, since for example $Prob(n_{\Delta_I,\Delta_O} = 4)$ depends on $\Delta_I \neq 0$, e.g. $Prob(n_{\Delta_I,\Delta_O} = 4) = 0$ if $\Delta_I = 0\text{x}F$ (i.e. $n_{0\text{x}F,\Delta_O} \neq 4\ \forall \Delta_O$) while $Prob(n_{\Delta_I,\Delta_O} = 4) = \frac{2}{15}$ if $\Delta_I = 0\text{x}A$ (two values of Δ_O satisfy $n_{0\text{x}A,\Delta_O} = 4$).

Assumptions on the S-Box. The fact that "the solutions of eq. (5.5) are uniform distributed for each $\Delta_I \neq 0$ and $\Delta_O \neq 0$" basically corresponds to work with an S-Box that satisfies the following properties: *(1st) it is "uniform" and (2nd) its $Var(n_{\Delta_I,\Delta_O})$ is as "low" as possible. This is close to being true if the S-Box is APN, or if the SBox is "close" to be APN.* Before going on, we remark that even if the variance $Var(n_{\Delta_I,\Delta_O})$ is related "in some sense" to DP_{max}, S-Boxes with equal DP_{max} can have very different variances. Moreover, the variance of an S-Box S_1 can be bigger than the corresponding variance of an S-Box S_2 even if DP_{max} of S_1 is lower than DP_{max} of S_2 (see Table 5.2 in Sect. 5.8 for examples).

Although much is known for (bijective) APN permutations in odd dimension, currently only little is known for the case of even dimension and what is known relies heavily on computer checking. In particular, there is no APN permutation of dimension 4 [LP07], while there is at least one APN permutation, up to equivalence, of dimension 6 - called the Dillon's permutation [BDMW10]. The question of finding an APN bijective (n, n)-function for even $n \geq 8$ is still open.

As a result, in the case of dimensions equal to a power of 2 (e.g. $\mathbb{F}_{2^4}$ or $\mathbb{F}_{2^8}$), *the only (known) S-Box that (approximately) matches the assumptions of the Theorem in dimensions 4 or 8 is the one generated by the multiplicative-inverse permutation unless affine equivalence relations*[7], as for example the AES S-Box, which is not APN but differentially 4-uniform [Nyb91] (e.g. note that the variance of the AES S-Box is 67 064/65 025 vs 64 004/65 025 of an APN S-Box). Our practical results on small-scale AES (for which the S-Box has the same property as the full-size AES one) are very close to the ones predicted by the previous Theorem.

We remark that even if the assumptions on the S-Box of Theorem 5 are restrictive, they match criteria used to design an S-Box which is strong against differential cryptanalysis. As a result, many ciphers in the literature are built using S-Boxes which (are close to) satisfy the assumptions of Theorem 5.

[6] In the case of a discrete probability distribution of a random variable X, the variance $Var(X) \equiv \sigma^2$ is defined as $\sigma^2 = E[(X - E[X])^2 = E[X^2] - E[X]^2$.

[7] Uniform differential property and DP_{max} of an S-Box $\mathcal{S}$ remain unchanged if affine transformations are applied in the domain or co-domain of $\mathcal{S}$. In more details, consider two S-Boxes $\mathcal{S}, \mathcal{S}' : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$. Let $A, B \in \mathbb{F}_2^{n \times n}$ be two invertible $n \times n$ matrices and $a, b \in \mathbb{F}_2^n$. $\mathcal{S}$ and $\mathcal{S}'$ are *affine equivalent* if and only if $\mathcal{S}'(x) = B \cdot [\mathcal{S}(A \cdot x + a)] + b$ $\forall x \in \mathbb{F}_2^n$.

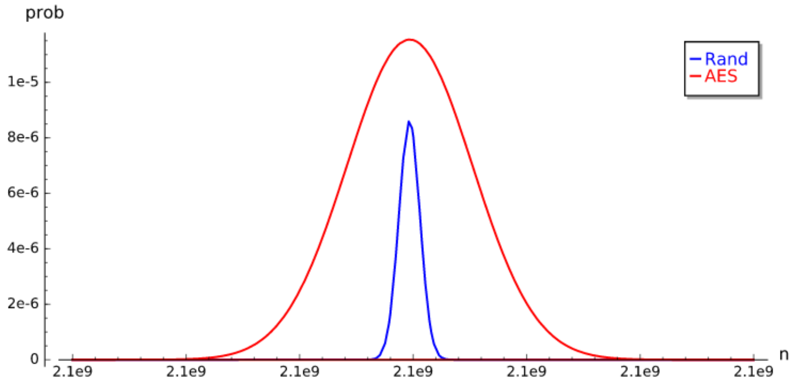


Figure 5.1.: Comparison between the theoretical probabilistic distribution of the number of collisions between 5-round AES (approximated by a normal distribution) and a random permutation.

Finally, we emphasize that if the S-Box does not satisfy the required properties related to the assumption of the Theorem, then the number of collisions can be different from the one previously given. To be more concrete, in Sect. 5.8 we provide several practical examples of the dependency of the number of collisions for small-scale AES-like ciphers w.r.t. the properties of the S-Box, and we provide theoretical argumentation to explain the influence of the S-Box. In the case in which the assumption about the S-Box is not fulfilled, it turns out that also the details of the MixColumns matrix can influence the average number of collisions.

5.2.3. Comparison between the Prob. Distribution of 5-round AES and of a Random Permutation

The previous results regarding the probabilistic distribution of the number of collisions for 5-round AES are not only of theoretically interest. As we are going to show, they can also be exploited in order to set up new truncated differential distinguishers for 5-round AES, which are independent of the secret-key. Thus, consider 2^{32} plaintexts in the same coset of $\mathcal{D}_i$, and the corresponding (cipher)texts generated by a random permutation $\Pi(\cdot)$ (or by an ideal cipher). *What is the probabilistic distribution of the number of different pairs of (cipher)texts generated by a random permutation* $\Pi(\cdot)$ *which are equal in one fixed anti-diagonal (assuming the final MixColumns operation is omitted)?*

Proposition 7. *Consider 2^{32} plaintexts p^i for $i = 0, 1, ..., 2^{32} - 1$ in a coset of a diagonal space $\mathcal{D}_i$, that is $\mathcal{D}_i \oplus a$ for $i \in \{0, 1, 2, 3\}$ and $a \in \mathcal{D}_i^{\perp}$, and the corresponding (cipher)texts generated by a random permutation Π, that is $c^i = \Pi(p^i)$. The probabilistic distribution of the number of different pairs of ciphertexts (c^i, c^j) with $c^i < c^j$ that belong to the same coset of $\mathcal{M}_J$ for $J \subseteq \{0, 1, 2, 3\}$ fixed with $|J| = 3$ (i.e. n collisions) is well approximated by a binomial distribution $\mathcal{B}(n, p)$, where*

$$n = \binom{2^{32}}{2} = 2^{31} \cdot (2^{32} - 1) \qquad \text{and} \qquad p = 2^{-32}.$$

The average number of collisions of such distribution is equal to $2^{31} - 0.5 = 2\,147\,483\,647.5$, while its variance is equal to $2\,147\,483\,647 \simeq 2^{31}$.

The main differences between the two distributions are the following:

- independently of the secret key, the average number of collisions is a (little) bigger for 5-round AES than for a random permutation (approximately 1 038.1 more collisions);
- independently of the secret key, the variance of the probabilistic distribution of the number of collisions is a (much) bigger for 5-round AES than for a random permutation (approximately of a factor 36).

To highlight this difference, Fig. 5.1 proposes a comparison between the probabilistic distribution of the number of collisions for the AES case in red - the probability to have $n \neq 8 \cdot n'$ collisions (i.e. where n is not a multiple of 8) is zero - and of the random case in blue. For simplicity and just for this figure, the probabilistic distribution for the AES case has been approximated by a normal distribution.

5.3. Proof of Theorem 5 – Mean of the Probabilistic Distribution of 5-round AES

In this section, we give a formal proof of Theorem 5, focusing on the values of p_3, p_{10}, p_{17} given there and on the average number of collisions for 5-round AES.

Reduction to the Middle Round

In order to prove the probability p_3, p_{10} and p_{17} given in Theorem 5, the idea is to prove an equivalent result on a single round.

Since each coset of a diagonal space is mapped into a mixed space after 2 rounds - see (4.2) - and since $Prob\big[t^1 \oplus t^2 \in \mathcal{D}_J \,|\, R^2(t^1) \oplus R(t^2) \in \mathcal{M}_J\big] = 1$, observe that for any $I, J \subseteq \{0,1,2,3\}$:

$$\mathcal{D}_I \oplus a' \xrightarrow[\text{prob. } 1]{R^2(\cdot)} \mathcal{M}_I \oplus b' \xrightarrow{R(\cdot)} \mathcal{D}_J \oplus a'' \xrightarrow[\text{prob. } 1]{R^2(\cdot)} \mathcal{M}_J \oplus b''.$$

Working on the middle round, the idea is to prove the following equivalent result. For simplicity, we limit ourselves to consider plaintexts in the same coset of $\mathcal{M}_0$ and to count the collisions in the same coset of a diagonal space $\mathcal{D}_{1,2,3}$ (the other cases are analogous). By definition of $\mathcal{M}_0$, if $p^1, p^2 \in \mathcal{M}_0 \oplus b'$ there exist $x^i, y^i, z^i, w^i \in \mathbb{F}_{2^8}$ for $i = 1, 2$ such that:

$$p^i = b' \oplus \begin{bmatrix} 2 \cdot x^i & y^i & z^i & 3 \cdot w^i \\ x^i & y^i & 3 \cdot z^i & 2 \cdot w^i \\ x^i & 3 \cdot y^i & 2 \cdot z^i & w^i \\ 3 \cdot x^i & 2 \cdot y^i & z^i & w^i \end{bmatrix}$$

where $2 \equiv$ 0x02 and $3 \equiv$ 0x03. According to Def. 13, we say that p^1 is "generated" by the generating variables (x^1, y^1, z^1, w^1) and that p^2 is "generated" by the generating variables (x^2, y^2, z^2, w^2) - we denote it by $p^i \equiv (x^i, y^i, z^i, w^i)$ for $i = 1, 2$.

The idea is to consider separately the following cases

- 3 variables are equal, e.g. $x^1 \neq x^2$ and $y^1 = y^2$, $z^1 = z^2$, $w^1 = w^2$;
- 2 variables are equal, e.g. $x^1 \neq x^2$,$y^1 \neq y^2$ and $z^1 = z^2$, $w^1 = w^2$;
- 1 variable is equal, e.g. $x^1 \neq x^2$, $y^1 \neq y^2$, $z^1 \neq z^2$ and $w^1 = w^2$;
- all variables are different, e.g. $x^1 \neq x^2$, $y^1 \neq y^2$, $z^1 \neq z^2$, $w^1 \neq w^2$.

In the first case - if 3 variables are equal (e.g. $y^1 = y^2$, $z^1 = z^2$ and $w^1 = w^2$), then $p^1 \oplus p^2 \in \mathcal{C}_k$ and $R(p^1) \oplus R(p^2) \in \mathcal{M}_k$ for a certain $k \in \{0,1,2,3\}$.Due to the "impossible differential trail" given in (4.6), it follows that $R(p^1) \oplus R(p^2) \notin \mathcal{D}_J$ for each J. Thus, in the following we limit ourselves to consider the case in which at least 2 generating variables are different.

Case: 2^{16} Texts with Two Equal Generating Variables

As a first case, we consider 2^{16} plaintexts in the same coset of $\mathcal{C}_{0,1} \cap \mathcal{M}_0$ (the other cases are equivalent). This corresponds to the case in which (at least) two generating variables are equal, e.g. $z^1 = z^2$ and $w^1 = w^2$.

Thus, consider two plaintexts p^1 generated by $(x^1, y^1, 0, 0)$ and p^2 generated by $(x^2, y^2, 0, 0)$ in $(\mathcal{C}_{0,1} \cap \mathcal{M}_0) \oplus b'$. By simple computation and by definition of the diagonal space $\mathcal{D}_{1,2,3}$, $R(p^1) \oplus R(p^2) \in \mathcal{D}_{1,2,3}$ if and only if the following four equations are satisfied

$$\begin{aligned}
(R(p^1) \oplus R(p^2))_{0,0} &= 2 \cdot \big[\text{S-Box}(2 \cdot x^1 \oplus a_{0,0}) \oplus \text{S-Box}(2 \cdot x^2 \oplus a_{0,0})\big] \oplus \\
&\oplus 3 \cdot \big[\text{S-Box}(y^1 \oplus a_{1,1}) \oplus \text{S-Box}(y^2 \oplus a_{1,1})\big] = 0,\\
(R(p^1) \oplus R(p^2))_{1,1} &= \text{S-Box}(3 \cdot x^1 \oplus a_{3,0}) \oplus \text{S-Box}(3 \cdot x^2 \oplus a_{3,0}) \oplus \\
&\oplus \text{S-Box}(y^1 \oplus a_{0,1}) \oplus \text{S-Box}(y^2 \oplus a_{0,1}) = 0,\\
(R(p^1) \oplus R(p^2))_{2,2} &= 2 \cdot \big[\text{S-Box}(x^1 \oplus a_{2,0}) \oplus \text{S-Box}(x^2 \oplus a_{2,0})\big] \oplus \\
&\oplus 3 \cdot \big[\text{S-Box}(2 \cdot y^1 \oplus a_{3,1}) \oplus \text{S-Box}(2 \cdot y^2 \oplus a_{3,1})\big] = 0,\\
(R(p^1) \oplus R(p^2))_{3,3} &= \text{S-Box}(x^1 \oplus a_{1,0}) \oplus \text{S-Box}(x^2 \oplus a_{1,0}) \oplus \\
&\oplus \text{S-Box}(3 \cdot y^1 \oplus a_{2,1}) \oplus \text{S-Box}(3 \cdot y^2 \oplus a_{2,1}) = 0.
\end{aligned}$$

Equivalently, four equations of the form

$$A \cdot \big[\text{S-Box}(B \cdot x^1 \oplus a) \oplus \text{S-Box}(B \cdot x^2 \oplus a)\big] \oplus C \cdot \big[\text{S-Box}(D \cdot y^1 \oplus c) \oplus \text{S-Box}(D \cdot y^2 \oplus c)\big] = 0 \quad (5.9)$$

must be satisfied, where A, B, C, D depend only on the MixColumns matrix, while a, c depend on the secret key and on the initial constant that defines the coset.

Number of Solutions of Each Equations. Consider one of these four equations. By simple observation, equation (5.9) is satisfied if and only if[8] the following system of equations is satisfied

$$\begin{aligned}
\text{S-Box}(\hat{x} \oplus \Delta_I) \oplus \text{S-Box}(\hat{x}) &= \Delta_O\\
\text{S-Box}(\hat{y} \oplus \Delta'_I) \oplus \text{S-Box}(\hat{y}) &= \Delta'_O\\
\Delta'_O &= C^{-1} \cdot A \cdot \Delta_O
\end{aligned} \quad (5.10)$$

for each value of Δ_O, where $\hat{x} = B \cdot x^1 \oplus a$, $\Delta_I = B \cdot (x^1 \oplus x^2)$, $\hat{y} = D \cdot y^1 \oplus c$ and $\Delta'_I = D \cdot (y^1 \oplus y^2)$.

What is the number of different (not null) solutions $[(x^1, y^1), (x^2, y^2)]$ *of eq. (5.9)?* Given Δ_O, each one of the first two equations of (5.10) admits $\frac{256}{255} \cdot 255 = 256$ different solutions $(\hat{x}, \Delta_I)$ (resp. $(\hat{y}, \Delta'_I)$) - note that there are 255 different values of $\Delta_I, \Delta'_I \neq 0$ and that the average number of solutions is 256/255. It follows that the number of different (not null) solutions $[(x^1, y^1), (x^2, y^2)]$ of eq. (5.9) - considering all the 255 possible values of Δ_O - is exactly equal to

$$\frac{1}{2} \cdot 255 \cdot \left(\frac{256}{255} \cdot 255\right)^2 = 255 \cdot 2^{15}$$

independently of the details of the S-Box. The factor 1/2 is due to the fact that we consider only different solutions, that is two solutions of the form $(p^1 \equiv (x^1, y^1), p^2 \equiv (x^2, y^2))$ and $(p^2 \equiv (x^1, y^1), p^1 \equiv (x^2, y^2))$ are considered equivalent. In other words, a solution $[(x^1, y^1), (x^2, y^2)]$ is considered to be valid if $x^2 \neq x^1$ and $y^1 < y^2$.

[8] Observe that the equality $\Delta'_O = A^{-1} \cdot C \oplus \Delta_O$ is well defined, since no coefficient of an MDS matrix $M \in \mathbb{F}_{2^b}^{4 \times 4}$ is equal to zero. Indeed, by definition, *a matrix M is MDS if and only if all square sub-matrices of M are of full rank.*

Probability of Common Solutions. Knowing the number of solutions of one eq. (5.9), *what is the number of common - different (not null) - solutions* $[(x^1, y^1), (x^2, y^2)]$ *of 4 equations of the form (5.9)?* We have just seen that each equation of the form (5.9) has exactly $255 \cdot 2^{15}$ different (not null) solutions $[(x^1, y^1), (x^2, y^2)]$. The probability that two equations admit the same solution (i.e. that $[(x^1, y^1), (x^2, y^2)]$ - solution of one equation - is equal to $[(\hat{x}^1, \hat{y}^1), (\hat{x}^2, \hat{y}^2)]$ - solution of another equation) is

$$(256 \cdot 255)^{-1} \cdot (255 \cdot 128)^{-1} = 255^{-2} \cdot 2^{-15}. \tag{5.11}$$

To explain this probability, the first term $(256 \cdot 255)^{-1}$ is due to the fact that $x^1 = \hat{x}^1$ with probability 256^{-1} while $x^2 = \hat{x}^2$ with probability 255^{-1}, since by assumption x^2 (resp. $\hat{x}^2$) can not be equal to x^1 (resp. $\hat{x}^1$). The second term $(128 \cdot 255)^{-1}$ is due to the assumption on the second variable, that is $y^1 < y^2$. To explain it[9], note that the possible number of pairs (y^1, y^2) with $y^1 < y^2$ is $\sum_{i=0}^{255} i = \frac{255 \cdot (255+1)}{2} = 255 \cdot 128$. It follows that y^1 and y^2 are equal to $\hat{y}^1$ and $\hat{y}^2$ with prob. $(128 \cdot 255)^{-1}$.

Total Number of Different - not null - Common Solutions. In conclusion, the average number of common - different (not null) - solutions $[(x^1, y^1), (x^2, y^2)]$ of 4 equations of the form (5.9) is given by

$$(255 \cdot 2^{15})^4 \cdot (255^{-2} \cdot 2^{-15})^3 = \frac{2^{15}}{255^2} \simeq 0.503929258 \simeq 2^{-1} + 2^{-7.992}.$$

For comparison, given plaintexts in the same coset of $\mathcal{M}_0 \cap \mathcal{C}_{0,1}$ and the corresponding ciphertexts generated by a random permutation, the average number of pairs of ciphertexts that belong to the same coset of $\mathcal{D}_J$ is approximately given by

$$\binom{2^{16}}{2} \cdot (2^{-8})^4 = \frac{2^{16} - 1}{2^{17}} \simeq 0.499992371 \simeq 2^{-1} - 2^{-17}.$$

Remark. We highlight that *probability (5.11) (strongly) depends on the assumptions that*

- the solutions of eq. (5.5) - so the numbers n_{Δ_I, Δ_O} - are uniform distributed for each $\Delta_I \neq 0$ and $\Delta_O \neq 0$;
- there is “no (obvious/non-trivial) relations” between the solutions of the studied system of four equations of the form (5.9); in other words, the four equations (5.9) must be independent/unrelated, in the sense that the solution of one equation must *not* be a solution of another one with probability different (bigger/smaller) than (5.11).

Let’s focus here on this second requirement. A relation among solutions of different equations *can* arise if some relations hold between the coefficients A, B, C, D of different equations of the form (5.9). Since these coefficients are the MixColumns coefficients, we deal with an MDS matrix (for which no linear relation among the rows/columns of any submatrix exist) in order to avoid this problem. More details about this are given in the following.

Case: 2^{24} Texts with One Equal Generating Variable

As second case, we consider 2^{24} plaintexts in the same coset of $\mathcal{C}_{0,1,2} \cap \mathcal{M}_0$ (the other cases are equivalent). This corresponds to the case in which (at least) one generating variable is equal, e.g. $w^1 = w^2$.

[9] As examples, if $y^1 = 0x0$ then y^2 can take 255 different values (all values except 0), if $y^1 = 0x1$ then y^2 can take 254 different values (all values except 0x0, 0x1) and so on - if $y^1 = d$ with $0 \leq d \leq 255$ then y^2 can take $255 - d$ different values.

As before, given two plaintexts $p^1, p^2 \in (\mathcal{C}_{0,1,2} \cap \mathcal{M}_0) \oplus b'$, they belong to the same coset of the diagonal space $\mathcal{D}_{1,2,3}$ if 4 equations of the form

$$\begin{aligned} A \cdot \big[\text{S-Box}(B \cdot x \oplus b) \oplus \text{S-Box}(B \cdot x' \oplus b)\big] \oplus \\ \oplus C \cdot \big[\text{S-Box}(D \cdot y \oplus d) \oplus \text{S-Box}(D \cdot y' \oplus d)\big] \oplus \\ \oplus E \cdot \big[\text{S-Box}(F \cdot z \oplus f) \oplus \text{S-Box}(F \cdot z' \oplus f)\big] = 0 \end{aligned} \tag{5.12}$$

are satisfied, where A, B, C, D, E, F depend only on the MixColumns matrix, while b, d, f depend on the secret key and on the initial constant that defines the coset. As before, each one of these equations is equivalent to a system of equations like (5.10), that is

$$\begin{aligned} \text{S-Box}(x \oplus \Delta_I) \oplus \text{S-Box}(x) = \Delta_O \\ \text{S-Box}(y \oplus \Delta'_I) \oplus \text{S-Box}(y) = \Delta'_O \\ \text{S-Box}(z \oplus \Delta''_I) \oplus \text{S-Box}(z) = \Delta''_O \end{aligned}$$

together with one of the two following conditions[10]:

1. $\Delta''_O = 0$ and $\Delta'_O = C^{-1} \cdot A \cdot \Delta_O$, or analogous (3 possibilities);
2. $\Delta_O, \Delta'_O, \Delta''_O \neq 0$ and $\Delta''_O = E^{-1} \cdot (A \cdot \Delta_O \oplus C \cdot \Delta'_O)$.

First Case. Since this first case is analogous to the case in which two generating variables are equal, we simply re-use the previous computation.

First of all note that if $\Delta''_O = 0$, then the third equation admits solutions if and only if $\Delta''_I = 0$. In other words, if $\Delta''_O = 0$, the only possible solutions of the third equation are $(z, \Delta''_I = 0)$ for each z. Using the same computation as before, the average number of (not null) common solutions for this first case is

$$\binom{3}{1} \cdot 256 \cdot \frac{2^{15}}{255^2} = \frac{2^{23}}{21\,675} \simeq 387.018.$$

Second Case. Consider now the case $\Delta_O, \Delta'_O, \Delta''_O \neq 0$ (i.e. $\Delta_I, \Delta'_I, \Delta''_I \neq 0$). First of all, note that $\Delta_O \neq 0$ can take 255 different values, while $\Delta'_O \neq 0$ can take only 254 different values. Indeed, it must be different from 0 and from $C^{-1} \cdot A \cdot \Delta_O$ (if $\Delta'_O = C^{-1} \cdot A \cdot \Delta_O$, then $\Delta''_O = 0$ which is excluded by assumption). Finally, the value of Δ''_O depends on Δ_O and Δ'_O.

Using the same argumentation as before, for each equation (5.12) the number of different solutions $[(x^1, y^1, z^1), (x^2, y^2, z^2)]$ - where $z^1 < z^2$ - is given by $\frac{1}{2} \cdot 255 \cdot 254 \cdot \big(255 \cdot \frac{256}{255}\big)^3 = 32\,385 \cdot 2^{24}$, while the probability that two equations of the form (5.12) have a common solution is given by $(256 \cdot 255)^{-2} \cdot (128 \cdot 255)^{-1} = 2^{-23} \cdot 255^{-3}$ *under the assumption (1st)* of uniform distribution of the solutions n_{Δ_I, Δ_O} of eq. (5.5) and *(2nd)* that there is "no (obvious/non-trivial) relation" between the solutions of the studied system of four equations of the form (5.12). It follows that for this second case we expect on average

$$(32\,385 \cdot 2^{24})^4 \cdot (2^{-23} \cdot 255^{-3})^3 = \frac{127^4 \cdot 2^{27}}{255^5} \simeq 32\,383.506$$

different - not null - common solutions for the 4 equations of the form (5.12).

[10] A solution of the first case can not be equal to a solution of the second case. Indeed, $\Delta''_O = 0$ implies $\Delta''_I = 0$ in the first case, while in the second one $\Delta_I, \Delta'_I, \Delta''_I \neq 0$.

Total Number of Different - not null - Common Solutions. By simple calculation, given plaintexts in the same coset of $\mathcal{C}_{0,1,2} \cap \mathcal{M}_0$, the average number of different pairs of ciphertexts that belong to the same coset of $\mathcal{D}_{1,2,3}$ is

$$32\,383.506 + 387.018 \simeq 32\,770.524 \simeq 2^{15} + 2^{1.336}$$

For comparison, if the ciphertexts are generated by a random permutation, the average number of different pairs of ciphertexts that belong to the same coset of $\mathcal{M}_J$ is approximately given by

$$\binom{2^{24}}{2} \cdot 2^{-32} \simeq 32\,767.998 \simeq 2^{15} - 2^{-9}$$

Generic Case: 2^{32} Texts

Finally, we consider 2^{32} plaintexts in the same coset of $\mathcal{M}_0$. This corresponds to the case in which all the generating variables are (potentially) different.

As before, given two plaintexts $p^1, p^2 \in \mathcal{M}_0 \oplus b'$, they belong to the same coset of $\mathcal{D}_{1,2,3}$ if four equations of the form

$$\begin{aligned} A \cdot \big[\text{S-Box}(B \cdot x \oplus b) \oplus \text{S-Box}(B \cdot x' \oplus b)\big] \oplus \\ \oplus C \cdot \big[\text{S-Box}(D \cdot y \oplus d) \oplus \text{S-Box}(D \cdot y' \oplus d)\big] \oplus \\ \oplus E \cdot \big[\text{S-Box}(F \cdot z \oplus f) \oplus \text{S-Box}(F \cdot z' \oplus f)\big] \oplus \\ \oplus G \cdot \big[\text{S-Box}(H \cdot w \oplus h) \oplus \text{S-Box}(H \cdot w' \oplus h)\big] = 0 \end{aligned} \tag{5.13}$$

are satisfied, where A, B, C, D, E, F, G, H depend only on the MixColumns matrix, while b, d, f, h depend on the secret key and on the constant that defined the initial coset. As before, each one of these equations is equivalent to a system of equations like (5.10), that is:

$$\begin{aligned} &\text{S-Box}(x \oplus \Delta_I) \oplus \text{S-Box}(x) = \Delta_O && \text{S-Box}(y \oplus \Delta_I') \oplus \text{S-Box}(y) = \Delta_O' \\ &\text{S-Box}(z \oplus \Delta_I'') \oplus \text{S-Box}(z) = \Delta_O'' && \text{S-Box}(w \oplus \Delta_I''') \oplus \text{S-Box}(w) = \Delta_O''' \end{aligned}$$

together with one of the following conditions

1. $\Delta_O''' = \Delta_O'' = 0$ and $\Delta_O' = C^{-1} \cdot A \cdot \Delta_O \neq 0$ or analogous (6 possibilities);
2. $\Delta_O''' = 0$, $\Delta_O, \Delta_O', \Delta_O'' \neq 0$ and $\Delta_O'' = E^{-1} \cdot (A \cdot \Delta_O \oplus C \cdot \Delta_O')$ or analogous, for a total of 4 possibilities;
3. $\Delta_O, \Delta_O', \Delta_O'', \Delta_O''' \neq 0$ and $\Delta_O''' = G^{-1} \cdot (A \cdot \Delta_O \oplus C \cdot \Delta_O' \oplus E \cdot \Delta_O'')$.

Since the first two conditions are analogous to the previous two cases already studied, we simply re-use the previous calculation.

First Case. In the case $\Delta_O''' = \Delta_O'' = 0$ and $\Delta_O' = C^{-1} \cdot A \cdot \Delta_O \neq 0$, the only possible solutions of the third and fourth equations are of the form $(z, \Delta_I'' = 0)$ and $(w, \Delta_I''' = 0)$ for each possible value of z and w. Using the same computation as before, the average number of (not null) common solutions for this case is

$$\binom{4}{2} \cdot 256^2 \cdot \frac{2^{15}}{255^2} = \frac{2^{32}}{21\,675} \simeq 198\,153.047. \tag{5.14}$$

This number can also be used in order to compute the probability p_{17} that texts with two equal (and two different) generating variables belong to the same coset of $\mathcal{D}_K$ after one round. By definition of probability:

$$p_{17} = \frac{1}{2^{17} \times n_{17}} \cdot \frac{2^{32}}{21\,675} = 2^{-32} + 2^{-37.98588}.$$

Second Case. Similarly, in the second case $\Delta_O^{'''} = 0$, $\Delta_O, \Delta_O^{'}, \Delta_O^{''} \neq 0$ and using the same computations as before, it follows that the average number of (not null) common solutions of this case is

$$\binom{4}{1} \cdot 256 \cdot \frac{127^4 \cdot 2^{27}}{255^5} = \frac{127^4 \cdot 2^{37}}{255^5} \simeq 33\,160\,710.047. \tag{5.15}$$

This number can also be used in order to compute the probability p_{10} that texts with one equal (and three different) generating variable(s) belong to the same coset of $\mathcal{D}_K$ after one round. By definition of probability:

$$p_{10} = \frac{1}{2^{10} \times n_{10}} \cdot \frac{127^4 \cdot 2^{37}}{255^5} = 2^{-32} - 2^{-45.98874}.$$

Third Case. We finally consider the case $\Delta_O, \Delta_O^{'}, \Delta_O^{''}, \Delta_O^{'''} \neq 0$. By simple computation, the number of different values that satisfy

$$\Delta_O^{'''} = G^{-1} \cdot (A \cdot \Delta_O \oplus C \cdot \Delta_O' \oplus E \cdot \Delta_O^{''}).$$

is given by $255^3 - (255 \cdot 254) = 16\,516\,605$. Indeed, the total number of $\Delta_O, \Delta_O^{'}, \Delta_O^{''} \neq 0$ is 255^3, while $255 \cdot 254$ is the total number of values $\Delta_O, \Delta_O^{'}, \Delta_O^{''} \neq 0$ for which $\Delta_O^{'''}$ is equal to zero (which is not possible since $\Delta_O^{'''} \neq 0$ by assumption). In more detail, *firstly* observe that for each value of Δ_O there is a value of $\Delta_O^{'}$ that satisfies $A \cdot \Delta_O = C \cdot \Delta_O'$. For this pair of values $(\Delta_O, \Delta_O' = C^{-1} \cdot A \cdot \Delta_O)$, the previous equation - which reduces to $\Delta_O^{'''} = G^{-1} \cdot E \cdot \Delta_O^{''}$ is always different from zero, since $\Delta_O^{''} \neq 0$. *Secondly*, for each one of the $255 \cdot 254$ values of the pair $(\Delta_O, \Delta_O' \neq C^{-1} \cdot A \cdot \Delta_O)$, there is only one value of $\Delta_O^{''}$ such that the previous equation is equal to zero.

As a result, the total number of different solutions $[(x^1, y^1, z^1, w^1), (x^2, y^2, z^2, w^2)]$ with $w^1 < w^2$ of each equation corresponding to (5.13) is

$$\frac{1}{2} \cdot 16\,516\,605 \cdot \left(255 \cdot \frac{256}{255}\right)^4 = 16\,516\,605 \cdot 2^{31}.$$

Since the probability that two solutions $[(x^1, y^1, z^1, w^1), (x^2, y^2, z^2, w^2)]$ and $[(\hat{x}^1, \hat{y}^1, \hat{z}^1, \hat{w}^1), (\hat{x}^2, \hat{y}^2, \hat{z}^2, \hat{w}^2)]$ are equal is $(255 \cdot 256)^{-3} \cdot (255 \cdot 128)^{-1} = 255^{-4} \cdot 2^{-31}$ – *under the assumption (1st)* of uniform distribution of the solutions n_{Δ_I, Δ_O} of eq. (5.5) and *(2nd)* that there is "no (obvious/non-trivial) relation" between the solutions of studied system of four equations 5.13, the average number of (non null) common solutions (with no equal generating variables) is

$$\left(16\,516\,605 \cdot 2^{31}\right)^4 \cdot (255^{-4} \cdot 2^{-31})^3 = \frac{64\,771^4 \cdot 2^{31}}{255^8} \simeq 2\,114\,125\,822.5 \tag{5.16}$$

This number can also be used in order to compute the probability p_3 that texts with no equal generating variable belong to the same coset of $\mathcal{D}_K$ after one round. By definition of probability:

$$p_3 = \frac{1}{2^3 \times n_3} \cdot \frac{64\,771^4 \cdot 2^{31}}{255^8} = 2^{-32} + 2^{-53.98306}.$$

Total Number of Different - not null - Common Solutions. By simple computation, given plaintexts in the same coset of $\mathcal{M}_0$, the number of different pairs of ciphertexts that belong to the same coset of $\mathcal{D}_{1,2,3}$ is

$$2\,114\,125\,822.5 + 33\,160\,710.047 + 198\,153.047 \simeq 2\,147\,484\,685.594 \simeq 2^{31} + 2^{10.02}.$$

Comparison with Random Permutation. For comparison, if the ciphertexts are randomly generated, the number of different pairs of ciphertexts that belong to the same coset of $\mathcal{M}_J$ is

$$\binom{2^{32}}{2} \cdot 2^{-32} \simeq 2\,147\,483\,647.5 = 2^{31} - 2^{-1}$$

In other words, on average there are

$$2\,147\,484\,685.594 - 2\,147\,483\,647.5 \simeq 1\,038.094$$

more collisions for 5-round AES than for a random permutation.

Finally, since the number of possible pairs of texts is $2^{31} \cdot (2^{32} - 1)$, the probability for the AES case that a couple of ciphertexts (c^1, c^2) satisfies $c^1 \oplus c^2 \in \mathcal{D}_J$ for $|J| = 3$ fixed is equal to

$$p_{AES} \simeq \frac{2\,147\,484\,685.594}{2^{31} \cdot (2^{32} - 1)} \simeq 2^{-32} + 2^{-52.9803} \tag{5.17}$$

versus 2^{-32} of the random case.

5.3.1. Remarks – On the Requirement that the MixColumns matrix is MDS

The assumption that the MixColumns matrix is MDS is *crucial* when computing the number of solutions of a system of 4 equations of the form (5.9) or (5.12) or (5.13) – remember that the coefficients $A, B, C, ...$ are the coefficients of the MixColumns matrix.

To give evidences of this, assume by contradiction that the matrix is not MDS, and focus on a system of equations e.g. of the form (5.9). First of all, if some coefficients of the MixColumns matrix are equal to zero, then some of such equations become trivial. E.g. if $C = 0$ then an equation of the form (5.9) is satisfied by $x^1 = x^2$ and by $y^1 \neq y^2$ (otherwise the two texts are equal). The problem arises since the arguments given for the case (5.9) hold only under the assumption $x^1 \neq x^2$ and $y^1 \neq y^2$. If the case $x^1 = x^2$ is admitted, the previous proof must modified accordingly – e.g. the number of solutions of an equation of the form (5.9) for $C = 0$ is $255 \cdot 2^{16}$, which differs from $255^2 \cdot 2^8$ given in the text, and the previous result must is not true anymore.

What happens if the MixColumns matrix is not MDS and it has no null coefficients? Consider a system of four equations of the form (5.9) or (5.12) or (5.13). Since the matrix is not MDS, then there exists a $r \times r$ submatrix (for $2 \leq r \leq 3$) whose determinant is equal to zero (this means that there exists a linear relation between the rows/columns of this matrix). This fact can have effects on the probability that the studied system of four equations has a common solution(s). In particular, *it can happen that the solutions of different equations of this system are not unrelated/independent, which is a crucial assumption exploited in the previous proof in order to compute the probability that different equations admit the same solution(s).* To better understand this fact, we show a concrete example in [GR18, App. D].

5.3.2. Generic Result on the Average Number of Collisions

For completeness, we generalize the previous result to the case in which the final mixed space is not fixed, that is we briefly discuss the case in which one considers the number of different pairs of ciphertexts that belong to the same coset of a mixed space $\mathcal{M}_K$ for an arbitrary $K \subset \{0, 1, 2, 3\}$ with $|K| = 3$.

Proposition 8 ([GR18])**.** *Consider an AES-like cipher that works with texts in $\mathbb{F}_{2^8}^{4\times 4}$ and for which the assumptions of Theorem 5 hold.*

Consider 2^{32} plaintexts p^i for $i = 0, 1, ..., 2^{32} - 1$ in a coset of a diagonal space $\mathcal{D}_k$, that is $\mathcal{D}_k \oplus a$ for $i \in \{0, 1, 2, 3\}$ and $a \in \mathcal{D}_k^{\perp}$, and the corresponding ciphertexts after 5 rounds, that is $c^i = R^5(p^i)$. The probability that a pair of ciphertexts (c^i, c^j) with $c^i < c^j$ belong to the same coset of $\mathcal{M}_K$ for $K \subseteq \{0, 1, 2, 3\}$ with $|K| = 3$ is equal to

- $2^{-32} + 2^{-52.9803}$ *if K is fixed to a (single) subset of {0,1,2,3} of cardinality 3;*
- $2^{-30} + 2^{-50.9803} - 2^{61.415} + ...$ *if K is not fixed (free to be equal to any subset of {0,1,2,3} of cardinality 3).*

A proof of this proposition can be found in [GR18, App. C]. For comparison, the same event has probability $2^{-30} - 2^{-61.415} + 2^{-94}$ in the case in which the ciphertexts are generated by a random permutation.

5.4. Proof of Theorem 5 – Variance – and of Lemma 5

Using the result just given, we finally compute the variance of the probabilistic distribution for 5-round AES given in Theorem 5, and the probability given in Lemma 5.

Let us recall that the probabilistic of 5-round AES is well described by

$$\text{5-AES} = 2^3 \cdot X_3 + 2^{10} \cdot X_{10} + 2^{17} \cdot X_{17}$$

where X_i are binomial distributions. The pairs of texts with no equal generating variables are represented by $2^3 \cdot X_3$, the pairs of texts with 1 equal generating variable are represented by $2^{10} \cdot X_{10}$ and finally the pairs of texts with 2 equal generating variables are represented by $2^{17} \cdot X_{17}$. We recall that given two plaintexts with three equal generating variables, then they can not belong to the same coset of $\mathcal{D}_J$ for $|J| \leq 3$ after one round.

5.4.1. Proof – Variance of the Prob. Distribution for 5-round AES

As we have just seen, note that all the previous cases (namely, X_3, X_{10} and X_{17}) are independent. In other words, the behavior of a pair of texts with v equal generating variables is independent of another pair with $\hat{v}$ equal generating variables where $\hat{v} \neq v$. One property of the variance is that, given x independent variables $X_1, ..., X_x$, the variance of $Y = X_1 + ... + X_x$ is given by $Var(Y) = Var(X_1) + ... + Var(X_x)$. It follows that the total variance of the probabilistic distribution for 5-round AES case is given by

$$\begin{aligned} Var(\text{5-AES}) &= Var(2^3 \cdot X_3) + Var(2^{10} \cdot X_{10}) + Var(2^{17} \cdot X_{17}) = \\ &= 2^6 \cdot Var(X_3) + 2^{20} \cdot Var(X_{10}) + 2^{34} \cdot Var(X_{17}), \end{aligned}$$

where $Var(\alpha \cdot X) = \alpha^2 \cdot Var(X)$.

Different Generating Variables. As first case, we consider the pairs of texts in which all the generating variables are different. The number of pairs with this property is $2^3 \cdot n_3 = 2^{31} \cdot (2^8 - 1)^4$. Again, the probability p_3 to have a collision for these pairs of texts is given by

$$p_3 = \underbrace{\frac{64\,771^4 \cdot 2^{31}}{255^8}}_{\text{see eq. (5.16)}} \cdot \frac{1}{8 \cdot n_3} = 2^{-32} + 2^{-53.98306}$$

using the results of the previous section.

As we have just seen, these pairs are not independent. By [GRR17], it is possible to divide them in $2^{31} \cdot (2^8 - 1)^4/8 = 2^{28} \cdot (2^8 - 1)^4 = n_3$ sets of 8 pairs such that for each set only two events can happen: *(1st)* all the pairs belong to the same coset of $\mathcal{D}_J$ after one round or *(2nd)* no one of them satisfies this property. Thus, the idea is to consider only independent pairs, i.e. one pair for each one of these sets, for a total of $2^{28} \cdot (2^8 - 1)^4$ pairs. Since these pairs are independent, the probabilistic distribution of the number of pairs that belong to the same coset of $\mathcal{D}_J$ after one round is given by a binomial distribution X_3 with mean value $\mu = n_3 \cdot p_3$ and variance $\sigma^2 = n_3 \cdot p_3 \cdot (1 - p_3) = 2^{28} \cdot (2^8 - 1)^4 \cdot (2^{-32} + 2^{-53.98306}) \cdot (1 - 2^{-32} - 2^{-53.98306}) \approx 264\,265\,727.751$, that is $Var(X_3) = 264\,265\,727.751$.

One Equal Generating Variable. As second case, we consider the pairs of texts in which all except one of the generating variables are different. The number of pairs with this property is $2^{10} \cdot n_{10} = 4 \cdot 2^{31} \cdot (2^8 - 1)^3$. Again, the probability p_{10} to have a collision for these pairs of texts is given by

$$p_{10} = \underbrace{\frac{127^4 \cdot 2^{37}}{255^5}}_{\text{see eq. (5.15)}} \cdot \frac{1}{2^{10} \cdot n_{10}} = 2^{-32} - 2^{-45.98874}$$

using the results of the previous section.

Working exactly as before, it is possible to divide them in $2^{33} \cdot (2^8 - 1)^3 / 2^{10} = 2^{23} \cdot (2^8 - 1)^3 = n_{10}$ sets of 2^{10} pairs. Considering only one pair for each one of these sets and since these pairs are independent, the probabilistic distribution of the number of pairs that belong to the same coset of $\mathcal{D}_J$ after one round is given by a binomial distribution X_{10} with mean value $\mu = n_{10} \cdot p_{10}$ and variance $\sigma^2 = n_{10} \cdot p_{10} \cdot (1 - p_{10}) = 2^{23} \cdot (2^8 - 1)^3 \cdot (2^{-32} - 2^{-45.98874}) \cdot (1 - 2^{-32} + 2^{-45.98874}) \approx 32\,383.506$, that is $Var(X_{10}) = 32\,383.506$.

Two Equal Generating Variables. As third case, we consider the case in which all except two of the generating variables are different, i.e. $v = 2$. The number of pairs with this property is $2^{17} \cdot n_{17} = 6 \cdot 2^{31} \cdot (2^8 - 1)^2$. Again, the probability p_{17} to have a collision for these pairs of texts is given by

$$p_{17} = \underbrace{\frac{2^{32}}{21\,675}}_{\text{see eq. (5.14)}} \cdot \frac{1}{2^{17} \cdot n_{17}} = 2^{-32} + 2^{-37.98588}$$

using the results of the previous section.

Working exactly as before, it is possible to divide them in $3 \cdot 2^{32} \cdot (2^8 - 1)^2 / 2^{17} = 3 \cdot 2^{15} \cdot (2^8 - 1)^2 = n_{17}$ sets of 2^{17} pairs. Considering only one pair for each one of these sets and since these pairs are independent, the probabilistic distribution of the number of pairs that belong to the same coset of $\mathcal{D}_J$ after one round is given by a binomial distribution X_{17} with mean value $\mu = n_{17} \cdot p_{17}$ and variance $\sigma^2 = n_{17} \cdot p_{17} \cdot (1 - p_{17}) = 3 \cdot 2^{15} \cdot (2^8 - 1)^2 \cdot (2^{-32} + 2^{-37.98588}) \cdot (1 - 2^{-32} - 2^{-37.98588}) \approx 1.51179$, that is $Var(X_{17}) = 1.51179$.

Final Result. By combining all previous results, it follows that

$$Var(\text{5-AES}) = 2^6 \cdot \underbrace{264\,265\,727.751}_{\simeq Var(X_3)} + 2^{20} \cdot \underbrace{32\,383.506}_{\simeq Var(X_{10})} + 2^{34} \cdot \underbrace{1.51179}_{\simeq Var(X_{17})} \simeq 2^{36.16118}.$$

5.4.2. Proof of Lemma 5

Consider 2^{32} plaintexts p^i for $i = 0, 1, ..., 2^{32} - 1$ in a coset of a diagonal space $\mathcal{D}_i$, that is $\mathcal{D}_i \oplus a$ for $i \in \{0, 1, 2, 3\}$ and $a \in \mathcal{D}_i^{\perp}$, and the corresponding ciphertexts after 5 rounds, that is $c^i = R^5(p^i)$. As last thing, we formally compute the probability to have $n \in \mathbb{N}$ different pairs of ciphertexts (c^i, c^j) with $c^i < c^j$ that belong to the same coset of $\mathcal{M}_J$ for $J \subseteq \{0, 1, 2, 3\}$ fixed with $|J| = 3$ (i.e. n collisions).

Given $n \in \mathbb{N}$, note that $Prob(\text{5-AES} = n) = 0$ if $n \neq 8 \cdot n'$ is not a multiple of 8 (due to the multiple-of-8 property). Thus, assume $n = 8 \cdot n'$ for $n' \in \mathbb{N}$:

$$\begin{aligned} &Prob\,(\text{5-AES} = n) := Prob\left([2^3 \cdot X_3 + 2^{10} \cdot X_{10} + 2^{17} \cdot X_{17}] = n\right) = \\ &= \sum_{n_3, n_{10}, n_{17}} Prob([2^3 \cdot X_3] = n_3) \times Prob([2^{10} \cdot X_{10}] = n_{10}) \times Prob([2^{17} \cdot X_{17}] = n_{17}) \times \\ &\quad \times Prob\left([2^3 \cdot X_3 + 2^{10} \cdot X_{10} + 2^{17} \cdot X_{17}] = n \middle| 2^3 \cdot X_3 = n_3, 2^{10} \cdot X_{10} = n_{10}, 2^{17} \cdot X_{17} = n_{17}\right) \end{aligned}$$

where remember that the distributions X_3, X_{10} and X_{17} are independent.

Since $Prob([2^i \cdot X_i] = n_i) = 0$ if $n_i \neq 2^i \cdot k_i$ for $i = 3, 10, 17$ and $k_i \in \mathbb{N}$ and since

$$Prob\left([2^3 \cdot X_3 + 2^{10} \cdot X_{10} + 2^{17} \cdot X_{17}] = n \middle| 2^3 \cdot X_3 = n_3, 2^{10} \cdot X_{10} = n_{10}, 2^{17} \cdot X_{17} = n_{17}\right) =$$
$$= \begin{cases} 1 & \text{if } n_3 + n_{10} + n_{17} = n \\ 0 & \text{otherwise} \end{cases}$$

it follows that $Prob\,(\text{5-AES} = n)$ is equal to

$$\sum_{k_3, k_{10}, k_{17} \in K_n} Prob(X_3 = k_3) \times Prob(X_{10} = k_{10}) \times Prob(X_{17} = k_{17})$$

where

$$K_n = \left\{(k_3, k_{10}, k_{17}) \in \mathbb{N} \times \mathbb{N} \times \mathbb{N} \,\middle|\, 0 \leq k_i \leq n_i \text{ and } 2^3 \cdot k_3 + 2^{10} \cdot k_{10} + 2^{17} \cdot k_{17} = n\right\}.$$

The probability given in Lemma 5 is finally obtained using the fact that X_i are binomial distributions:

$$Prob(X_i = x) = \binom{n_i}{x} \cdot (p_i)^x \cdot (1 - p_i)^{n_i - x},$$

where n_i and p_i for $i = 3, 10, 17$ are given in Theorem 5.

5.5. Relation among Multiple-of-8, Mean and Variance

Before going on, we discuss the relations among the multiple-of-8 property, the fact that the average number of collisions - the mean in the following - is higher for AES than for a random permutation and the fact that the variance of the number of collisions is (much) higher for AES than for a random permutation. As we are going to highlight, *there is no "obvious relation" between the multiple-of-8 property and the result on the mean*, while the multiple-of-8 property and the result on the variance are strictly related.

Relation between Multiple-of-8 Property and the Mean

As we just said, the multiple-of-8 property and the result on the mean are unrelated/independent:

- the fact that the number of collisions is always a multiple of 8 for AES does *not* imply that such number is on average bigger/equal/smaller for AES;
- the fact that the number of collisions is on average higher for AES does *not* imply that it is a multiple-of-8.

To give concrete examples, we practically computed the average number of collisions for 4-bit AES when the AES S-Box is replaced by the S-Box of other ciphers — see Table 5.2 in Sect. 5.8. *In all cases, the number of collisions always satisfies the multiple of 8 property. Instead, depending on the S-Box details, it's possible that the number of collisions is higher or smaller (or potentially equal) for AES than for a random permutation* (more details in the following). This supports the arguments that these two properties are independent.

The independence of these two results is also motivated by the fact that the reasons for which these two properties hold are completely different and independent. E.g. let's focus on the corresponding proofs[11]. Even if both proofs focus on the middle round $\mathcal{M}_I \oplus a \xrightarrow{R(\cdot)} \mathcal{D}_J \oplus b$, their goals are very different:

[11] The fact that both the two proofs are divided in cases is a choice made in order to simplify their understanding.

Multiple-of-8: given two texts in $t^1, t^2 \in \mathcal{M}_I \oplus a$, the goal is to show that *other pairs of texts* $s^1, s^2 \in \mathcal{M}_I \oplus a$ *defined by considering all possible combinations of the generating variables of* t^1 *and* t^2 satisfy the following equivalence

$$R(t^1) \oplus R(t^2) = R(s^1) \oplus R(s^2).$$

More concretely, assume $|I| = 1$ and let (x^i, y^i, z^i, w^i) be the *generating variables* of t^i for $i = 1, 2$, that is $t^i \equiv \langle x^i, y^i, z^i, w^i \rangle \in \mathcal{M}_I \oplus a$. As proved in Sect. 5.1, $R(t^1) \oplus R(t^2) = R(s^1) \oplus R(s^2)$ for each pairs of texts $s^1, s^2 \in \mathcal{M}_I \oplus a$ defined by a different combinations of the generating variables.

To summarize, *the multiple-of-8 property depends on the facts that (1st) the XOR-sum is commutative and (2nd) the S-Box works independently on each generating variable.*

Mean: in this case, the goal is to count the average number of pairs of texts t^1 and t^2 that satisfy $R(t^1) \oplus R(t^2) \in \mathcal{D}_J$, which can be re-written as a system of four equations of the form

$$\forall k, j \text{ s.t. } \big[(k - j) \bmod 4\big] \notin J : \qquad (R(t^1) \oplus R(t^2))_{k,l} = 0,$$

As we have seen in Sect. 5.3, this corresponds to count the *average number of common solutions of systems of four equations* of the form

$$\begin{gathered}
\text{S-Box}(x^2 \equiv x^1 \oplus \Delta_I^x) \oplus \text{S-Box}(x^1) = \Delta_O^x \quad \text{S-Box}(y^2 \equiv y^1 \oplus \Delta_I^y) \oplus \text{S-Box}(y^1) = \Delta_O^y \\
\text{S-Box}(z^2 \equiv z^1 \oplus \Delta_I^z) \oplus \text{S-Box}(z^1) = \Delta_O^z \quad \text{S-Box}(w^2 \equiv w^1 \oplus \Delta_I^w) \oplus \text{S-Box}(w^1) = \Delta_O^w \\
A \cdot \Delta_O^x \oplus B \cdot \Delta_O^y \oplus C \cdot \Delta_O^z \oplus D \cdot \Delta_O^w = 0
\end{gathered}$$

where as before (x^i, y^i, z^i, w^i) are the *generating variables* of t^i for $i = 1, 2$, and the constants A, B, C, D depend on the MixColumns matrix. By contrast to the multiple-of-8 property, *the final result for the mean is obtained by probabilistic considerations, under precise assumptions on the S-Box.* To the best of our knowledge, *this is the first time that a similar approach is used in the literature.*

We also remark that the first proof is independent of the details of the S-Box, while the second one depends on them. Finally, while the first proof is deterministic (everything is deterministic - probability plays no role) and the multiple-of-8 property holds with prob. 1, the second proof is probabilistic.

Relation between Multiple-of-8 Property and the Variance

Vice-versa, as we have just seen, the multiple-of-8 property and the variance are strictly related. Roughly speaking, due to the multiple-of-8 property (i.e. due the fact that the pairs of texts are not independent), the probabilistic distribution of the number of collisions Y can be rewritten as $Y = \alpha \times X$ for a constant $\alpha > 1$, where X is the probabilistic distribution of the number of collisions for the *independent/unrelated* pairs of texts (see Theorem 5 for details). Since

$$Var(Y) = Var(\alpha \times X) = \alpha^2 \times Var(X),$$

it turns out that the variance for 5-round AES is higher than the corresponding variance of a random permutation (note instead that the mean value does not have this property, since $\mathbb{E}[Y] = C \times \mathbb{E}[X]$).

5.6. Practical Results on AES

We have practically verified the mean and the variance for 5-round AES given above – see Theorem 5 – using a C/C++ implementation[12]. In particular, we have verified the mean value on a small-scale AES as proposed in [CMR05], and the variance value both on full-size and on the small-scale AES. We limit ourselves to recall that the AES small-scale S-Box is defined in the same way as the full-size one and that it has the same properties as the full-size one, with the only exception that each word is composed of 8 bits for full-size AES and of 4 bits for the small-scale one. We emphasize that our verification on the small-scale variant of AES is strong evidence for it to hold for the full-size AES, since the strategy used to theoretically compute such probabilities is independent of the fact that each word of AES is of 4 or 8 bits.

5.6.1. 5-round AES defined over $(\mathbb{F}_{2^n})^{4\times 4}$

For completeness, we propose a generic result about the average number of collisions for 5-round AES defined over $\mathbb{F}_{2^n}^{4\times 4}$.

Theorem 6 ([GR18]). *Consider an AES-like cipher that works with texts in $\mathbb{F}_{2^n}^{4\times 4}$, s.t. the Mix-Columns matrix is an MDS matrix and s.t. the solutions of eq. (5.5) are uniformly distributed for each input/output difference $\Delta_I \neq 0$ and $\Delta_O \neq 0$.*

Consider 2^{4n} plaintexts p^i for $i = 0, 1, ..., 2^{4n} - 1$ in a coset of a diagonal space $\mathcal{D}_i$, that is $\mathcal{D}_i \oplus a$ for $i \in \{0, 1, 2, 3\}$ and $a \in \mathcal{D}_i^{\perp}$, and the corresponding ciphertexts after 5 rounds, that is $c^i = R^5(p^i)$. The distribution probability 5-AES of the number of different pairs of ciphertexts (c^i, c^j) with $c^i < c^j$ that belong to the same coset of $\mathcal{M}_J$ for $J \subseteq \{0, 1, 2, 3\}$ fixed with $|J| = 3$ is described by

$$5\text{-}AES = 2^3 \times X_3 + 2^{n+2} \times X_{n+2} + 2^{2n+1} \times X_{2n+1}$$

where

$$\forall i = 3, n+2, 3n+1: \qquad X_i \sim \mathcal{B}(n_i, p_i)$$

are binomial distributions s.t.

$$n_3 = 2^{4n-4} \cdot (2^n - 1)^4 \qquad p_3 = \frac{(2^{2n} - 3 \cdot 2^n + 3)^4}{(2^n - 1)^{12}};$$

$$n_{n+2} = 2^{3n-1} \cdot (2^n - 1)^3 \qquad p_{n+2} = \frac{(2^{n-1} - 1)^4 \cdot 2^4}{(2^n - 1)^8};$$

$$n_{2n+1} = 3 \cdot 2^{2n-1} \cdot (2^n - 1)^2 \qquad p_{2n+1} = \frac{1}{(2^n - 1)^4}.$$

The average number of different pairs of ciphertexts (c^i, c^j) with $c^i \leq c^j$ for $i \neq j$ that belong to the same coset of $\mathcal{M}_K$ for $K \subseteq \{0, 1, 2, 3\}$ fixed with $|K| = 3$ is equal to

$$\frac{2^{4n-1} \cdot (2^{2n} - 3 \cdot 2^n + 3)^4}{(2^n - 1)^8} + \frac{(2^{n-1} - 1)^4 \cdot 2^{4n+5}}{(2^n - 1)^5} + 3 \cdot \frac{2^{4n}}{(2^n - 1)^2},$$

while the variance of such distribution is given by

$$\frac{2^{4n+2} \cdot (2^{2n} - 3 \cdot 2^n + 3)^4}{(2^n - 1)^8} + \frac{(2^{n-1} - 1)^4 \cdot 2^{6n+9}}{(2^n - 1)^5} + \frac{3 \cdot 2^{6n+1}}{(2^n - 1)^2}$$

A complete proof of this Theorem – equivalent to the one just given for the case $(\mathbb{F}_{2^8})^{4\times 4}$ – can be found in [GR18].

[12] The source codes of the distinguishers are available at `https://github.com/Krypto-iaik/Distinguisher_5RoundAES`

5.6.2. Practical Verification on 4-bit AES

Theoretical Results To compare the practical values with the theoretical ones, we first re-propose Theorem 5 for the case of small-scale AES.

Lemma 6. *Consider an AES-like cipher that works with texts in $\mathbb{F}_{2^4}^{4\times 4}$ and for which the assumptions of Theorem 5 hold.*

Consider 2^{16} plaintexts p^i for $i = 0, 1, ..., 2^{16} - 1$ in a coset of a diagonal space $\mathcal{D}_k$, that is $\mathcal{D}_k \oplus a$ for $i \in \{0, 1, 2, 3\}$ and $a \in \mathcal{D}_k^\perp$, and the corresponding ciphertexts after 5 rounds, that is $c^i = R^5(p^i)$. The distribution probability of the number of different pairs of ciphertexts (c^i, c^j) with $c^i \leq c^j$ for $i \neq j$ that belong to the same coset of $\mathcal{M}_K$ for $K \subseteq \{0, 1, 2, 3\}$ fixed with $|K| = 3$ is well approximated by a Normal Distribution with mean value $\mu = 32\,847.124$ and variance $\sigma^2 = 982\,466.615$ (or equivalently, standard deviation $\sigma = 991.195$).

For comparison, in the case in which the ciphertexts are generated by a random permutation, the distribution probability of the number of collisions is well approximated by a normal distribution with mean value $\mu = 32\,767.5$ and variance $\sigma^2 = 32\,767$ (or equivalently, standard deviation $\sigma = 181.017$).

Practical Results. In order to test our results, we took the variance over 320 initial cosets for full-size AES, while we took the average number of collisions and the variance over respectively $125\,000 \simeq 2^{17}$ and over 100 initial cosets for the small-scale one.

The variance results for full-size AES[13] are given in the following

$$\sigma_T^2 = 76\,842\,293\,834.905 \simeq 2^{36.161} \qquad \sigma_P^2 = 73\,288\,132\,411.36 \simeq 2^{36.093}$$

where the subscript $\cdot_T$ denotes the theoretical value and the subscript $\cdot_P$ the practical one.

Our practical results for small-scale AES regarding the mean - denoted by μ - are

$$\mu_{AES}^T = 32\,847.124 \qquad \mu_{rand}^T = 32\,767.5$$
$$\mu_{AES}^P = 32\,848.57 \qquad \mu_{rand}^P = 32\,768.2$$

while our practical results for small-scale AES regarding the variance - denoted by σ^2 - are

$$\sigma_{AES}^T = 991.195 \qquad \sigma_{rand}^T = 181.02$$
$$\sigma_{AES}^P = 1023.06 \qquad \sigma_{rand}^P = 182.42$$

where as before the superscript $\cdot^T$ the theoretical values and the superscript $\cdot^P$ the practical ones.

Fig. 5.2 highlights the difference between the *practical* probabilistic distribution of the number of collisions for small-scale AES and for a permutation drawn at random.

Remark – Mean, Mode and Skewness. *About Fig. 5.2, it is important not to confuse the mean and the mode.* In particular, consider a random variable X with a finite number of outcomes $x_0, x_1, ..., x_n$ occurring with probabilities $p_0, p_1, ..., p_n$ respectively (where $\sum_i p_i = 1$):

mean: the mean - or *expected value* - of such a random variable X is defined as $\mu = \mathbb{E}[X] = \sum_{i=0}^n p_i \times x_i$;

mode: the mode of a set of data values is the value - if exists - that appears most often, that is $\text{mode}(X) = \big\{x_i \in X \,|\, \forall j = 0, ..., n, j \neq i : p_i > p_j\big\}$.

[13] We remark that one would need more than one year of computation on our cluster to test the distinguisher based on the mean with its $\approx 2^{16}$ initial cosets.

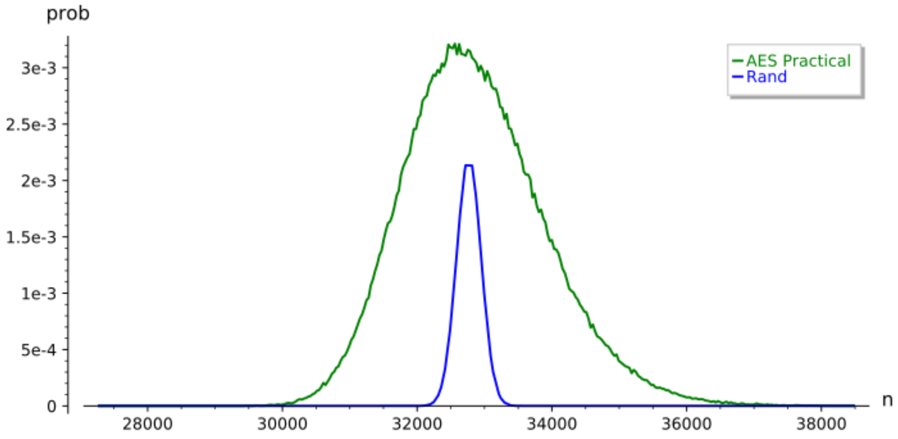


Figure 5.2.: Comparison between the *practical* probabilistic distributions of the number of collisions of small-scale 5-round AES and of a random permutation.

In our case, consider the probabilistic distribution for 5-round AES: the mean of such distribution is approximately equal to $\mu_{AES}^{P} = 32\,848.57$, while the mode is approximately equal to 32 560. For the random case, the mean and the mode are approximately equal (the distribution is approximately symmetric).

It is important to have in mind that *for skewed* (i.e. asymmetric) *distributions, the mean is not necessarily the same as the most likely value, i.e. the mode.* In particular, the mean and the mode coincide only in the case in which the skewness is equal to zero, which is the case of e.g. a normal distribution (which is always symmetric).

The skewness is a parameter that measures the asymmetry of the probabilistic distribution of a real-valued random variable about its mean. The skewness value can be positive or negative, or undefined. In particular, referring to Figure 5.3[14], the skew is negative if the left tail is longer (i.e. the mass of the distribution is concentrated on the right of the figure), while it is positive if the right tail is longer (i.e. the mass of the distribution is concentrated on the left of the figure).

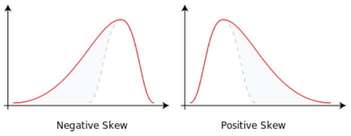


Figure 5.3.: Examples of negative and positive skew.

The skewness of a random variable X is the third standardized moment γ, defined as:

$$\gamma = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^3\right]$$

where $\mathbb{E}[\cdot]$ is the mean value operator, $\mu \equiv \mathbb{E}[X]$ the mean value and $\sigma^2 \equiv Var(X)$ the variance[15]. For the particular case of a binomial distribution $\mathcal{B}(n,p)$, the skewness is given by

$$\gamma = \frac{1 - 2\cdot p}{\sqrt{n\cdot p\cdot (1-p)}}, \tag{5.18}$$

which is close to zero if $p \approx 1/2$ or if $n \cdot p \gg 1$.

[14] Figure re-printed from Wikipedia `https://en.wikipedia.org/wiki/Skewness`

[15] For a sample of n values, an estimator z for the skewness is given by $z = \{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{X})^3\}/\{\frac{1}{n}\sum_{i=1}^{n}[(x_i-\bar{X})^2])^{3/2}\}$ where $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

The Probabilistic Distribution for 5-round AES is *not* Symmetric – A Distinguisher based on the Skewness? Interestingly, it is possible to observe an *asymmetry in the (small-scale) 5-round AES distribution.*

By Fig. 5.2-5.4, it is possible to observe that small-scale 5-round AES distribution has positive skew, while the skew of the random distribution is approximately equal to zero.

We practically computed these values both for full-size AES and for small-scale one using 2^9 initial cosets, and we got the following results:

$$\gamma^{AES} \simeq 0.43786 \qquad\qquad \gamma^{AES}_{\text{small-scale}} \simeq 0.4687$$

while we got that the skew of a random permutation is close to 0 (hence, the probabilistic distribution of a random permutation is well described by a normal one).

It follows that also the skew can be used to set up a distinguisher. We leave the open problem to theoretically compute these numbers, both for small-scale AES and full-size AES, and to set up a corresponding distinguisher.

5.7. Truncated Differential Distinguishers for 5-round AES

5.7.1. Truncated Differential Distinguisher based on the Variance

The fact that *the variance of the AES case is different from the one of the random case independently of the secret-key* can be exploited to set up a new secret-key distinguisher for 5-round AES.

The idea is very simple. Given n different cosets of a diagonal space $\mathcal{D}_i$, one counts the number of different pairs of ciphertexts that belong to the same coset of $\mathcal{M}_J$ for each J with $|J| = 3$. Then, one computes the variance: by previous result, the highest one corresponds to the AES case.

We practically tested this distinguisher on a small-scale AES. Since the ratio between the variances for full-size AES permutation and for a random permutation is similar to the same ratio in the case of small-scale AES, that is

$$\frac{276\,469.4}{46\,340.95} \approx 6 \approx \frac{991.195}{181.02},$$

we conjecture that the results obtained for the small-scale AES are applicable as well to full-size AES.

By practical tests (the following probability of success have been computed over 2 500 tests)[16] on small-scale AES:

- a single initial coset of $\mathcal{D}_i$ allows to distinguish small-scale AES from a random permutation with prob. 98%;
- two initial cosets of $\mathcal{D}_i$ allows to distinguish small-scale AES from a random permutation with prob. 99.9%.

Note that for each initial coset of $\mathcal{D}_i$, it is possible to compute the average number of collisions with respect to four different anti-diagonals, or equivalently four different subspaces $\mathcal{M}_J$. Moreover, we emphasize that *the goal of this distinguisher is not to compute the exact value of the variance for the two cases, but to distinguish them.* In other words, the distinguisher works if the variance for AES is bigger than the one of a random permutation, even if it does not return the exact value of the two variances. Due to the big gap between the two cases, 2 initial cosets of $\mathcal{D}_i$ are sufficient for this goal (even if they are not sufficient to compute the exact value of the two variances).

As a result, one can distinguish the two cases using $n \geq 2$ initial cosets, or in other words 2 initial cosets are largely sufficient to "accurately" compute the variance for the AES case and the

[16] Given a set of $n \gg 1$ equally likely values, an *unbiased* estimator for the variance is given by $Var(X) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{X})^2$ where $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

random one. Due to the relation between small-scale AES and full-size AES previously discussed, we conjecture that the same number of initial cosets is sufficient to distinguish (full-size) AES from a random permutation (using this distinguisher based on the variance). However, just to have more confidence, we choose an arbitrary value of 4 initial cosets in order to set up the distinguisher, for a data cost of $2^2 \cdot 2^{32} = 2^{34}$ chosen plaintexts distributed in 4 initial cosets of $\mathcal{D}_j$. The computational cost is well approximated by the cost to compute the number of collisions. Using e.g. Algorithm 1, the cost is well approximated by $4 \cdot 2^2 \cdot 3 \cdot 2^{32} \simeq 2^{37.6}$ table look-ups, that is approximately 2^{31} five-round encryptions.

5.7.2. Useful Approximation for the Prob. Distribution for 5-round AES

In order to propose a truncated differential distinguisher based on the mean, we first need an approximation of the probabilistic distribution for 5-round AES given in Theorem 5 – Lemma 5. The approximation given in the following turns out to be (very) useful in all applications where the skewness does *not* play a crucial role, that is in all applications which are (almost) independent of the bias in the skew.

As given in Theorem 5, the probabilistic distribution for 5-round AES is well described by

$$\text{5-AES} = 2^3 \times \mathcal{B}(n_3, p_3) + 2^{10} \times \mathcal{B}(n_{10}, p_{10}) + 2^{17} \times \mathcal{B}(n_{17}, p_{17})$$

where $\mathcal{B}(n,p)$ are binomial approximation. Since $n \gg 1$ and $p \ll 1$, a first possibility would be to approximate the binomial distributions by Poisson ones, that is $\mathcal{B}(n,p) \approx \mathcal{P}(\lambda)$ where $\lambda = n \cdot p$. On the other hand, given the probabilistic distribution $2^3 \cdot \mathcal{P}(n_3 \cdot p_3) + 2^{10} \times \mathcal{P}(n_{10} \cdot p_{10}) + 2^{17} \times \mathcal{P}(n_{17} \cdot p_{17})$, it seems hard to derive a closed "simple" formula which describes the probability to have n collisions in the ciphertexts[17]. The same occurs using a Gamma distribution (the "continuous counterpart" of the Poisson one).

Another possibility would be to approximate the binomial distributions using the corresponding normal ones (see Sect. 4.6.1 for more details on this). The De Moivre-Laplace Theorem claims that the normal distribution is a good approximation of the binomial one *if* the skewness of the binomial distribution – given in (5.18) – is close to zero. In our case, $\mathcal{B}(n_3, p_3)$ and $\mathcal{B}(n_{10}, p_{10})$ can be well approximated by a normal distribution, since their skewness are close to zero[18]. Unfortunately, this is not the case of X_{17}:

$$skew(X_3) \approx 2^{-14} \qquad skew(X_{10}) \approx 2^{-7.5} \qquad skew(X_{17}) \approx 0.813 \approx 2^{-0.3}.$$

On the other hand, the number of pairs represented by X_{17} (that is, the pairs of texts with two equal generating variables) is very small compared to the number of all possible pairs of texts, precisely $\frac{3 \cdot 2^{15} \cdot (2^8-1)^2}{2^{31} \cdot (2^{32}-1)} \approx 2^{-30.4}$. For this reason – and with the only goal to set up the truncated diff. distinguisher in the following, we make use of this approximation.

Finally, we exploit the fact that the sum of normally distributed random variables is also normally distributed, that is if $X \sim N(\mu_X, \sigma_X^2)$ and $Y \sim N(\mu_Y, \sigma_Y^2)$, then $Z = X + Y \sim N(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2)$, in order to get an approximation for the probabilistic distribution of 5-round AES.

Corollary 1. *Consider an AES-like cipher that works with texts in $\mathbb{F}_{2^8}^{4\times 4}$ and for which the assumptions of Theorem 5 hold.*

Consider 2^{32} plaintexts p^i for $i = 0, 1, ..., 2^{32}-1$ in a coset of a diagonal space $\mathcal{D}_i$, that is $\mathcal{D}_i \oplus a$ for $i \in \{0,1,2,3\}$ and $a \in \mathcal{D}_i^{\perp}$, and the corresponding ciphertexts after 5 rounds, that is $c^i = R^5(p^i)$. The probabilistic distribution of the number of different pairs of ciphertexts (c^i, c^j) with $c^i < c^j$ that belong to the same coset of $\mathcal{M}_J$ for $J \subseteq \{0,1,2,3\}$ fixed with $|J| = 3$ (i.e. n collisions) is approximated by

[17] While it is well known that $\mathcal{P}(\lambda_1) + \mathcal{P}(\lambda_2) = \mathcal{P}(\lambda_1 + \lambda_2)$, to the best of our knowledge there is no closed formula for the case $a_1 \cdot \mathcal{P}(\lambda_1) + a_2 \cdot \mathcal{P}(\lambda_2)$ for $a_1 \neq a_2$.

[18] Note that $skew(\alpha \cdot X) = sign(\alpha) \cdot skew(X)$ where $sign(\alpha) = -1$ if $\alpha < 0$, and 1 otherwise.

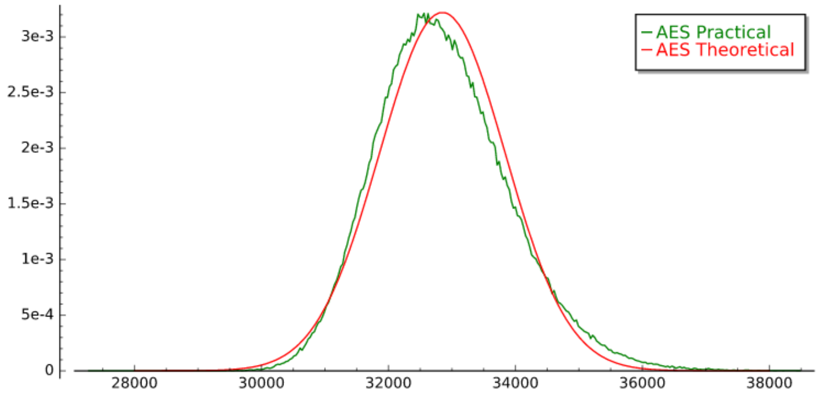


Figure 5.4.: Comparison between the probabilistic distribution of the number of collisions between theoretical small-scale 5-round AES (approximated by a normal distribution) and a practical one.

a normal distribution $\mathcal{N}(\mu, \sigma^2)$, where the mean value is equal to $\mu = 2\,147\,484\,685.6 = 2^{32} + 1\,037.6$ and standard deviation is equal to $\sigma = 277\,204.426$.

Roughly speaking, the distribution of the number of collisions for the AES case is approximated by $8 \times X$, where X is a normal distribution with mean value and variance as given in Corollary 1. In more details, the (discrete) probability to have $n \in \mathbb{N}$ collisions is given by:

$$Prob(n \mid \mu, \sigma^2) = \begin{cases} 0 & \text{if } n \bmod 8 \neq 0 \\ \underbrace{\frac{8}{\sqrt{2 \cdot \pi \cdot \sigma^2}} \cdot e^{-\frac{(n-\mu)^2}{2 \cdot \sigma^2}}}_{\sim 8 \times N(\mu, \sigma^2)} & \text{otherwise} \end{cases}$$

A comparison between the real probabilistic distribution for small-scale 5-round AES and the (theoretical) one approximated by a normal distribution is proposed in Fig. 5.4. As expected, the variance of the two distributions are equal, while the main difference is the skewness (the skewness of a normal distribution is zero, while the skewness of the probabilistic distribution for 5-round AES is approximately 0.4). Moreover, as already pointed out before, it is important not to confuse the mean with the mode – see Sect. 5.6.2 for details.

Finally, a brief explanation about the factor 8 in the probability $Prob(n \mid \mu, \sigma^2)$. Let $Prob(n)$ be the probability - just defined - to have n collisions for 5-round AES. Since $Prob(n \neq 8 \cdot n') = 0$ (i.e. the probability to have n collisions is zero if n is not a multiple of 8), we highlight that *the factor 8 guarantees that the total probability is equal to 1:*

$$\sum_n Prob(n) = \sum_{n = 8 \cdot n'} \frac{8}{\sqrt{2 \cdot \pi \cdot \sigma^2}} \cdot e^{-\frac{(n-\mu)^2}{2 \cdot \sigma^2}} = \sum_{n'} \frac{1}{\sqrt{2 \cdot \pi \cdot (\sigma/8)^2}} \cdot e^{-\frac{(n' - (\mu/8))^2}{2 \cdot (\sigma/8)^2}} = 1.$$

5.7.3. Truncated Differential Distinguisher based on the Mean

Another distinguisher that can be set up for 5-round AES is based on the previous result about the mean, that is the fact that the average number of collisions in $\mathcal{M}_J$ for each J with $|J| = 3$ is a little bigger for AES than for a random permutation.

As discussed in the previous section, the number of collisions for 5-round AES and for the random permutation are well described by normal distributions. Moreover, to derive concrete numbers for our distinguisher, we can simply consider the difference of the two distributions, which is again a normal distribution. That is, given $X \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $Y \sim \mathcal{N}(\mu_2, \sigma_2^2)$, then $X - Y \sim \mathcal{N}(\mu, \sigma^2) = \mathcal{N}(\mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2)$. Indeed, note that to distinguish the two cases, it is sufficient to guarantee that the average number of pairs that satisfy the required property for the random case is smaller than for AES. As a result, the mean μ and the variance σ^2 of the difference between the AES and the random distributions are

$$\mu = |\mu_{AES} - \mu_{rand}| = n \cdot |p_{AES} - p_{rand}|$$
$$\sigma^2 = \sigma^2_{rand} + \sigma^2_{AES} = n \cdot \left[p_{rand} \cdot (1 - p_{rand}) + 35.593 \cdot p_{AES} \cdot (1 - p_{AES})\right]$$

Since the probability density of the normal distribution is $f(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, it follows that

$$prob = \int_{-\infty}^{0} \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}} \mathrm{d}x = \int_{-\infty}^{-\mu/\sigma} \frac{1}{\sqrt{2\pi}}\, e^{-\frac{x^2}{2}} \mathrm{d}x = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{-\mu}{\sigma\sqrt{2}}\right)\right],$$

where erf(x) is the error function, defined as the probability of a random variable with normal distribution of mean 0 and variance 1/2 falling in the range $[-x, x]$. We emphasize that the integral is computed in the range $(-\infty, 0]$ since we are interested only in the case in which the average number of pairs with the required property in the random case is smaller than in the AES case.

In order to have a probability of success bigger than *prob*, n has to satisfy

$$n > \frac{2 \cdot [p_{rand} \cdot (1 - p_{rand}) + 35.593 \cdot p_{AES} \cdot (1 - p_{AES})]}{(p_{rand} - p_{AES})^2} \cdot \left[\operatorname{erfinv}\big(2 \cdot prob - 1\big)\right]^2.$$

where erfinv(x) is the inverse error function.

For the case $p_{rand}, p_{AES} \ll 1$, a good approximation of n is given by[19]

$$n > \frac{73.186 \cdot \max(p_{rand},\, p_{AES})}{(p_{rand} - p_{AES})^2} \cdot \left[\operatorname{erfinv}\big(2 \cdot prob - 1\big)\right]^2. \tag{5.19}$$

It follows that in order to have a probability of success bigger than 95%, the number of pairs must satisfy $n \geq 2^{78.374}$, since $p_{rand} \approx p_{AES} \approx 2^{-30}$ and $|p_{rand} - p_{AES}| \approx 2^{-50.98}$. Since each coset of $\mathcal{D}_k$ contains 2^{32} different texts and approximately 2^{63} different pairs, this means that the distinguisher requires $2^{15.374}$ different cosets for a data cost of $2^{47.374}$ chosen plaintexts.

Remark. We emphasize that the formula given in (5.19) is equivalent to the one proposed by Matsui in [Mat93; Mat94] for the linear cryptanalysis case, which has been rigorously studied in the literature (e.g. in [BJV04] and in [SB02; Sel08]). As we have seen, in linear cryptanalysis one has to construct "good" linear equations relating plaintext, ciphertext and key bits. In order to find the secret key, the idea is to exploit the fact that such linear approximation holds with probability 1/2 for a wrong key, while they hold with probability $1/2 \pm \varepsilon$ for the right key. Exploiting this (usually small) difference between the two probabilities, one can discover the secret key. Note that also these events can be described by binomial variables, that is $\mathcal{B}(n, 1/2)$ for a wrongly guessed key and $\mathcal{B}(n, 1/2 \pm \varepsilon)$ for the right guessed key, where n is the number of texts used. Our case is completely equivalent, since the probability p_{AES} for the AES case is related to the probability p_{rand} for the random case by $p_{AES} = p_{rand} \pm \varepsilon$, for a small difference ε.

Practical Results on small-scale AES. Since the previous result has been obtained under the assumption that the distribution of AES is well approximated by a normal distribution, we practically tested the probability of success of such distinguisher on a small-scale AES. Using the

[19] Observe: $p_{rand} \cdot (1 - p_{rand}) + 35.593 \cdot p_{AES} \cdot (1 - p_{AES}) < p_{rand} + 35.593 \cdot p_{AES} < 36.593 \cdot \max(p_{rand}, p_{AES})$.

same computation as before, it turns out that for small-scale AES (denoted by AES* – where $\mu_{AES^\star} = n \cdot p_{AES^\star}$ and $\sigma^2_{AES^\star} = 29.983 \cdot n \cdot p_{AES^\star} \cdot (1 - p_{AES^\star}))$ one needs

$$n > \frac{59.965 \cdot \max(p_{rand}, p_{AES^\star})}{(p_{rand} - p_{AES^\star})^2} \cdot \Big[\text{erfinv}\big(2 \cdot prob - 1\big)\Big]^2.$$

different pairs of texts to set up the distinguisher with prob. *prob*. In order to have a probability of success higher than 95%, since $p_{rand} \approx p_{AES^\star} \approx 2^{-14}$ and $|p_{rand} - p_{AES^\star}| \approx 2^{-22.68485}$, it follows that the number of pairs must satisfy $n \geq 2^{37.48}$. Since each coset of $\mathcal{D}_i$ contains 2^{16} different texts and approximately 2^{31} different pairs, this means that the distinguisher requires $2^{6.48} \simeq 90$ cosets for a data cost of $2^{22.48}$ chosen plaintexts.

By practical tests on small-scale AES:

- 90 initial cosets of $\mathcal{D}_i$ allows to distinguish small-scale AES from a random permutation with prob. 92% (close to 95% used before);
- 180 initial cosets of $\mathcal{D}_i$ allows to distinguish small-scale AES from a random permutation with prob. 98.5%;
- 270 initial cosets of $\mathcal{D}_i$ allows to distinguish small-scale AES from a random permutation with prob. 99.9%;

where the previous probability of success have been computed over 2 500 tests. The fact that the probability of success is a little lower than expected is well justified by used of an approximation of the probabilistic distribution of AES. Due to these results, due to the similarity between small-scale AES and AES (e.g. the value of the skewness is similar for these two cases – see Sect. 5.6.2) and just to have more confidence, we choose an arbitrary value of $3 \cdot 2^{15.375} = 2^{16.96}$ initial cosets in order to set up the distinguisher for AES, for a data cost of $2^{16.96} \cdot 2^{32} = 2^{48.96}$ chosen plaintexts distributed in $2^{16.96}$ initial cosets of $\mathcal{D}_j$.

The Computational Cost. We have just seen that $2^{48.96}$ chosen plaintexts (i.e. $2^{16.96}$ cosets of $\mathcal{D}_I$ with $|I| = 1$) are sufficient to distinguish a random permutation from 5 rounds of AES, simply counting the number of pairs of ciphertexts that belong to the same coset of $\mathcal{M}_J$ for $|J| = 3$ and using the fact that this number is bigger for AES. Here we give an estimation of the computational cost of the distinguisher, which is (approximately) given by the cost to count the number of collisions. Using Algorithm 1, the total computational cost can be well approximated by $2^{52.6}$ table look-ups, or equivalently 2^{46} five-round encryptions of AES (using the approximation 20 table look-ups $\approx$ 1 round of AES).

5.8. Open Problem - 5-round Truncated Distinguisher for Generic AES-like Ciphers

To summarize, we have presented a new truncated property for 5-round AES-like ciphers in the case in which "the solutions of equation (5.5) are uniformly distributed for each input/output difference $\Delta_I \neq 0$ and $\Delta_O \neq 0$", which is close to being true if the S-Box is APN, or if the SBox is "close" to be APN. Even if no S-Box (completely) satisfies this assumption in $\mathbb{F}_{2^4}$ or $\mathbb{F}_{2^8}$, the theoretically result of Theorem 5 matches the practical result obtained for the AES S-Box, which approximately satisfies the assumptions of such Theorem (as discussed in Sect. 5.2.2). Thus, natural questions arise: *What happens when the AES S-Box is changed with an S-Box that does not satisfy (at all) the assumptions of Theorem 5? Is it possible to naturally extend our results to any general case?*

We have studied this problem working on small-scale AES, and by practical results the answer to the second question seems to be negative. In other words, our theory does not extend naturally to generic S-Box, but it should be modified depending on the particular properties/details of the S-Box function.

Preliminary Considerations and Practical Results

To summarize, in Sect. 5.3 we used the fact that the average number of solutions x of the differential S-Box$(x \oplus \Delta_I)\oplus$ S-Box$(x) = \Delta_O$ is 256/255 (for 8-bit AES). This is independent of the details of the S-Box.

Now, consider the probabilistic distribution of the number of solutions x of the previous equations for non-zero Δ_I, Δ_O. Obviously, the mean of such probabilistic distribution is 256/255. Roughly speaking, in Sect. 5.3 we computed our result by assuming that the variance of such distribution is zero. Obviously, this can not be the case. *Since the variance of such distribution depends on the details of the S-Box, we expect that our theoretical results match the practical ones when one works with an S-Box that minimizes such variance*, which happens – in the best case – when one works with an APN S-Box. However, the variance of such distribution when using the AES S-Box is very close to the variance of such distribution when using an APN S-Box ("Variance APN = 64004/65025" versus "Variance AES = 67064/65025").

Here we start an analysis in order to better understand which properties of the S-Box play a crucial role when computing the average number of collisions for 5-round AES. In more details, we did several practical tests by counting the average number of collisions in the case in which the AES S-Box is replaced with other S-Box permutations present in the literature - PRINCE [BCG+12], MIDORI [BBI+15], KLEIN [GNL11], PRESENT [BKL+07], RECTANGLE [ZBL+15], NOEKEON [DPAR00] and PRIDE [ADK+14] - and with some "toy" S-Boxes. For our tests, given 2^{16} plaintexts in the same coset of $\mathcal{D}_i$, we counted the average number of collision in the same coset of $\mathcal{M}_J$ for J fixed with $|J| = 3$ and we computed the mean. The obtained results are listed in Table 5.2, where we also highlight some properties of the used S-Box (definitions and differential spectrum of the used S-Boxes are given in [GR18, App. I]) and the difference between the number of collisions found by experiments and the theoretical number 32 847.124 under the assumptions of Theorem 6 (while the average number of collisions for a random permutation is 32 767.5). For each AES-like cipher, we used $125\,000 \simeq 2^{17}$ different initial cosets (values given in the table are the average ones) - new keys are generated at random for each test.

We emphasize that, while all these AES-like ciphers satisfy the multiple-of-8 property, for some of them the average number of collisions is bigger than the case of a random permutation (e.g. AES S-Box), while for others it is smaller (e.g. Toy-12 S-Box). This supports again the *independence* of the multiple-of-8 property from the fact that the average number of collisions is bigger for 5-round AES.

Observations and (possible) Explanation

As expected, *the (absolute) difference between the found number of collisions and the theoretical one seems to increase when the variance (of the S-Box) increases, while it seems to be independent of the maximum differential probability DP_{max}*. Moreover, the difference between the theoretical number of collisions (given under the assumptions of Theorem 5 - the number of solutions n_{Δ_I,Δ_O} of equation (5.5) are uniform distributed) and the practical one is minimum when the S-Box almost satisfies the assumption of Theorem 5 - e.g. the AES S-Box.

To explain these results, we must refer to the proof of Theorem 5 given in Sect. 5.3. The idea is to consider a system of 4 equations of the generic form (5.13), and to look for common solutions. In the case in which the solutions (in particular, the number of solutions n_{Δ_I,Δ_O}) of equation (5.5) are uniformly distributed, the probability that a possible solution satisfies all the 4 equations of the system is well approximated by $(255^{-4} \cdot 2^{-31})^3$, as explained in the proof of Sect. 5.3. This allows to (theoretical) predict the average number of common solutions, and so of collisions. Instead, in the

Table 5.2.: In the following table, we provide the results of our practical tests about the number of different pairs of ciphertexts that belong to the same coset of $\mathcal{M}_J$ for J fixed with $|J| = 3$ when *the AES S-Box is replaced by the S-Box of other ciphers*. Together with the number of collisions, we provide the most relevant properties of the S-Box ("Var" denotes the variance of the probabilistic distribution that describes the number of solutions of the eq. S-Box$(x \oplus \Delta_I) \oplus$ S-Box$(x) = \Delta_O$) and the "Difference" between the practical and the theoretical number (= 32 847.124) of collisions - under the assumptions of Theorem 5.

AES-like Cipher	**Numb. Collisions**	**Diff.**	$2^4 \times$ **DP$_{\max}$**	**Var**	**Unif. Diff.**
AES [CMR05]	32 848.6	+1.45	4	344/225	✓
KLEIN [GNL11]	32 849.8	+2.7	4	344/225	
MIDORI SB_1 [BBI+15]	32 843.0	−4.1	4	344/225	
PRINCE [BCG+12]	32 852.7	+5.6	4	344/225	
Toy-6 [GR18]	32 840.1	−7.1	6	392/225	
RECTANGLE [ZBL+15]	32 861.2	+14.0	4	416/225	
NOEKEON [DPAR00]	32 878.7	+31.6	4	416/225	
MIDORI SB_0 [BBI+15]	32 882.8	+35.7	4	416/225	
PRESENT [BKL+07]	32 886.3	+39.2	4	416/225	
PRIDE [ADK+14]	32 806.6	−40.5	4	416/225	
Toy-8 [GR18]	32 815.7	−31.5	8	464/225	
Toy-10 [GR18]	32 919.0	+71.9	10	864/225	
Toy-12 [GR18]	32 684.1	−163.0	12	896/225	

case in which the solutions (in particular, the number of solutions n_{Δ_I,Δ_O}) of equation (5.5) are not uniform distributed (e.g. if the variance of the S-Box is not "low"), then the probability to have a common solution is in general different from the one just given. As a result, the number of solutions of a system of equations like (5.13) can be bigger or smaller w.r.t. the one given in Theorem 5 (and the difference can be also non-negligible). It follows that the number of collisions is influenced by the details of the S-Box (as expected). *As future work, an open problem is to theoretically prove this conjecture about the link between the average number of collisions and the variance of the S-Box, and to theoretically derive the numbers given in Table 5.2.*

What about the distinguisher based on the variance? To compute the value of the variance, we have exploited the "multiple-of-8" property, the properties of the Variance (if X is a random variable and a a scalar, then $Var(a \cdot X) = a^2 \cdot Var(X)$) and the probability p_{AES} that - given a pair of plaintexts in $\mathcal{D}_i$ - two ciphertexts belong to the same coset of $\mathcal{M}_J$ after 5-round. This probability p_{AES} (5.17) depends on the details of the S-Box, as we have just seen. It follows that also the value of the variance depends on it. On the other hand, we found by practical tests that *the value of the variance changes much less than the corresponding value of the mean* when the S-Box changes. In general, the value of the variance is "almost" independent of the details of the S-Box. Moreover, since the variance for an AES-like cipher is much bigger than the one of a random permutation, the proposed distinguisher works even if the value of the variance is (a little) different than the one given in Theorem 5.

Future Open Problems. As a result, while we provide a theoretical explanation (besides practical verifications) of our results, an *open problem* is to adapt our theoretical argumentations to the cases in which the S-Box does not satisfy the assumptions of Theorem 5. As first step, we conjecture an explanation of our results in this last case, but more research in that sense must be done.

MixColumns Dependence

Until now, we have focused only on the details of the S-Box. *How does the average number of collisions depend on the details of the MixColumns matrix?*

MDS Matrix: "Good" vs "Bad" S-Box. We start by focusing on the case in which the MixColumns matrix is MDS, and then we briefly discuss the other cases. *If the S-Box satisfies the assumptions of Theorem 5, then the average number of collisions is (almost) independent of the MixColumns matrix details. Instead, if the S-Box does not satisfy the previous requirement, this number depends also on the details of the MixColumns matrix.* In particular, in this last case the solutions (and the corresponding number n_{Δ_I,Δ_O}) of equation (5.5) are not uniform distributed with respect to $\Delta_I \neq 0$ and $\Delta_O \neq 0$, and so the number of solutions of a system of 4 equations of the generic form (5.13) depends both on the details of the S-Box and of the linear layer. Indeed, remember that a system of equations of the generic form (5.13) depends on the coefficients of the MixColumns matrix, and so also the fact that a common solution exists.

To give a practical example, consider the (circulant) MixColumns matrix defined as

$$MC = circ(0x01, 0x03, 0x02, 0x02),$$

that is the AES MixColumns matrix where 0x01 is replaced by 0x02 and vice-versa. We got that the number of collisions in the case of AES S-Box is 32 850.32, while in the case of PRESENT S-Box is 32 872.95. Thus, a difference in the MixColumns matrix implies almost no difference for the AES S-Box case (on average, there are +1.75 collisions for this new MDS matrix), while an higher difference occurs for the PRESENT S-Box case (on average, there are −13.37 collisions for this new MDS matrix). As we have just said, this is due to the fact that the probability that a system of 4 equations of the generic form (5.13) admits a common solution both on the details of the S-Box and of the linear layer, in the case in which the S-Box is not "good" (w.r.t. assumptions of Theorem 5). Similar results can be obtained using different MDS MixColumns matrices.

Non-MDS Matrix. Finally, if the AES MixColumns matrix is replaced by an "almost MDS" one (which does not satisfy the assumptions of Theorem 5), then the number of collisions can be different with respect to the one predicted by Theorem 5 also in the case of "good" S-Box. As example, using the Midori matrix

$$MC_{Midori} = circ(0x00, 0x01, 0x01, 0x01)$$

and the AES S-Box, the number of collisions after 5-round is on average 31 883.27 (instead of a theoretical number of 32 847.124). The same occurs also using a MixColumns matrix which is not MDS and for which all coefficients are different than zero. E.g. using the matrix $circ(0x02, 0x01, 0x01, 0x01)$ and the AES S-Box, the number of collisions after 5 rounds is on average 33 377.93 (instead of a theoretical number of 32 847.124).

5.9. Key-Recovery Attacks on 5-round AES

Finally, we propose several (new) attacks on 5-round AES that exploit the secret-key distinguishers just proposed here revisited on 4-round AES.

Why Not an Attack on 6-round AES? To give an overview, consider the following aspect. To construct the proposed distinguishers, one consider a full coset of a subspace $\mathcal{D}_i$ for $i \in \{0, 1, 2, 3\}$ - that is, a set of 2^{32} plaintexts with one active diagonal, and exploits properties that are related to the number of ciphertexts that belong to a subspace $\mathcal{M}_J$. In order to exploit directly these distinguishers, one can guess the final key, decrypt the ciphertexts, counts the number of collisions in the same coset of $\mathcal{M}_J$ and exploits one of the proposed properties. However, since a coset of $\mathcal{M}_J$ is mapped into the full space, it seems hard to check this property one round before without guessing the entire key. Similar considerations can be done if the guessed key is the initial one. It follows that it is rather hard to set up an attack different than a brute force one that exploits directly the proposed 5-round distinguishers - this open problem is left for *future work*. For comparison, note that such a problem does not arise for the other distinguishers up to 4-round AES (e.g. the impossible differential or the integral ones), for which it is sufficient to guess only part of the secret key in order to verify if the required property is satisfied or not.

Thus, we consider round-reduced distinguishers on 4-round to propose new key-recovery attacks.

Idea of the Attacks. Instead of working with 2^{32} plaintexts with one active diagonal, we consider 2^{24} texts with three active bytes in the same column, e.g. a coset of $\mathcal{D}_{0,2,3} \cap \mathcal{C}_0$. As we are going to show, the properties just presented hold after 4-round in the same way. To set up the attacks, the idea is to extend the distinguishers at the beginning and to partially guess the initial key. In more details, consider 2^{32} plaintexts in $\mathcal{D}_0 \oplus a$. After one round, they are mapped into a coset of $\mathcal{C}_0$ with prob. 1. However, the way in which they are divided in cosets of $\mathcal{D}_{0,2,3} \cap \mathcal{C}_0$ depends on the guessed key

$$2^{32} \text{ plaintexts in } \mathcal{D}_0 \oplus b \xrightarrow[\text{(partially) key-guess}]{R(\cdot)} 2^{24} \text{ texts in } \mathcal{D}_{0,2,3} \cap \mathcal{C}_0 \oplus a \xrightarrow{R^4(\cdot)} ...$$

$$... \xrightarrow{R(\cdot)} 2^{24} \text{ texts in } \mathcal{D}_{0,2,3} \cap \mathcal{C}_0 \oplus a \xrightarrow{R^4(\cdot)} \text{ distinguisher property.}$$

We exploit this fact to set up new key-recovery attacks on 5-round AES.

In more details, the attacks that we are going to present are based on the following properties:

- the number of collisions is a multiple of 2/4/8;
- the average number of collisions is (a little) bigger for AES than for a random permutation;
- the variance of the number of collisions is higher for AES than for a random permutation.

In the following, we first present the generic strategy to set up these attacks (which is common for all the previous cases), and then we give all the details.

5.9.1. Generic Strategy

In order to exploit one of the previous properties, the idea is the following. Consider 2^{24} texts in the same coset of $\mathcal{D}_I \cap \mathcal{C}_j$ for $|j| = 1$ and $|I| = 2$ or $|I| = 3$, e.g.

$$\mathcal{D}_{0,2,3} \cap \mathcal{C}_0 \oplus a \equiv \begin{bmatrix} A & C & C & C \\ A & C & C & C \\ A & C & C & C \\ C & C & C & C \end{bmatrix},$$

and the corresponding ciphertexts after 4-round. The idea of the attack is to guess 4 bytes of the key (i.e. the j-th diagonal), to partially compute 1-round decryption of $\mathcal{D}_I \cap \mathcal{C}_j \oplus a$ and to ask for the corresponding ciphertexts after 5-round. Exploiting one of the previous properties that hold on the ciphertexts only if the guessed key is the right one, it is possible to filter wrong keys and to find the right one. In particular, this is due to the fact that if the guessed key is not the right one, the behavior is the same of a random permutation - *Wrong-Key Randomization Hypothesis.*

In more details, consider $2^{24 \cdot n}$ texts in n cosets of $\mathcal{D}_I \cap \mathcal{C}_j$. The idea is to compute 1-round decryption with respect to a guessed key and ask for the corresponding ciphertexts. The following properties holds

- *the number of collisions is always a multiple of 2 if* $|I| = 2$ *and of 4 if* $|I| = 3$ *for the right key*, while it can assume any value for a wrong guessed key;
- *the average number of collisions in the same coset of* $\mathcal{M}_J$ *for* J *fixed with* $|J| = 3$ *is approximately equal to* 32 770.524 *for the right key*, while it is approximately 32 767.998 for a wrong guessed key;
- *the variance of the number of collisions is approximately equal to* $2^{17.8}$ *for the right key*, while it is approximately 2^{15} for a wrong guessed key.

Note that if $n \le 2^8$ initial cosets are sufficient to set up the attack, then the data cost of this step is at most of 2^{32} chosen plaintexts in the same coset of $\mathcal{D}_i$, since $\mathcal{D}_I \cap \mathcal{C}_j \oplus b \subseteq \mathcal{C}_j \oplus b = R(\mathcal{D}_i \oplus a)$. When one diagonal of the key is found, the other ones can be found using the same strategy or by brute force.

Wrong-Key Randomization Hypothesis. One assumption of the attack is the wrong-key randomization hypothesis. This hypothesis states that *decrypting one or several rounds with a wrong key guess creates a function that behaves like a random one.* This assumption is very common and used for classical/truncated/impossible differentials key-recovery attacks.

For this reason, we limit ourselves to show that it holds also in our case. Consider 2^{24} texts t^i in a coset of $\mathcal{D}_{0,2,3} \cap \mathcal{C}_0$ for $i = 0, ..., 2^{24} - 1$, and let k the secret subkey and $\hat{k}$ the guessed key. The decryption under the guessed key $\hat{k}$ is simply given by:

$$R_{\hat{k}}^{-1}(t^i) = \hat{k} \oplus \text{ S-Box}^{-1} \circ SR^{-1} \circ MC^{-1}(t^i).$$

To implement the attack, one asks the corresponding ciphertexts after 5-round (with respect to the right key k). By simple computation, after one round

$$R_k \circ R_{\hat{k}}^{-1}(t^i) = MC \circ SR \circ \text{ S-Box}\left[\hat{k} \oplus k \oplus \text{ S-Box}^{-1} \circ SR^{-1} \circ MC^{-1}\left(t^i\right)\right].$$

Thus, if $\hat{k} = k$, then $R_k \circ R_{\hat{k}}^{-1}(t^i) = t^i$ for each i, and the distinguisher property holds. On the other hand, if $\hat{k} \neq k$, then $R_k \circ R_{\hat{k}}^{-1}(t^i) \neq t^i$ for each i since the S-Box is a non-linear operation. It follows that $\{R_k \circ R_{\hat{k}}^{-1}(t^i)\}_i$ do not belong to the same coset of $\mathcal{D}_{0,2,3} \cap \mathcal{C}_0$, and the distinguisher property does not work. In this case, the behavior is the same of a random permutation, and the attacker can filter wrong keys.

Implementation Strategy. In the following we give the details of the attack. We highlight that in all cases the attacker has to count the number of collisions in the same coset of $\mathcal{M}_J$ in order to filter wrong keys. Even if it is possible to use the strategy proposed in Algorithm 1, another strategy is more competitive in this case.

The basic idea is to re-order the texts with respect to a partial order $\preceq$ and to work only on consecutive ordered texts. In particular, since our goal is to check if two texts belong to the same coset of $\mathcal{M}_J$ for $|J| = 3$, the idea is to re-order the texts using a particular numerical order which depends by J. Then, given a set of ordered texts, the idea is to work only on two consecutive elements in order to count the total number of collisions. In other words, given ordered ciphertexts, one can work only on approximately 2^{32} different pairs (composed of consecutive elements with respect to the used order) instead of 2^{63} for each coset of $\mathcal{D}_I$.

In order to implement such strategy, we first define the following *partial order* $\preceq$:

Definition 14. *Let* $I \subset \{0, 1, 2, 3\}$ *with* $|I| = 3$ *and let* $l \in \{0, 1, 2, 3\} \setminus I$. *Let* $t^1, t^2 \in \mathbb{F}_{2^8}^{4\times 4}$ *with* $t^1 \neq t^2$. *The text* t^1 *is less or equal than the text* t^2 *with respect to the partial order* $\preceq$ *(i.e.* $t^1 \preceq t^2$*) if and only if one of the two following conditions is satisfied (the indexes are taken modulo 4):*

- *there exists* $j \in \{0,1,2,3\}$ *such that for all* $i < j$*:*

$$MC^{-1}(t^1)_{i,l-i} = MC^{-1}(t^2)_{i,l-i} \qquad \textit{and} \qquad MC^{-1}(t^1)_{j,l-j} < MC^{-1}(t^2)_{j,l-j};$$

- *given* $\leq$ *defined as in Def. 12, for all* $i = 0, ..., 3$*:*

$$MC^{-1}(t^1)_{i,l-i} = MC^{-1}(t^2)_{i,l-i} \qquad \textit{and} \qquad MC^{-1}(t^1) \leq MC^{-1}(t^2).$$

Thus, as first step, one must re-order the 2^{32} ciphertexts of each coset with respect to the partial order relationship $\preceq$ defined before.

After the re-ordering process, in order to count the number of pairs of texts that belong to the same coset of $\mathcal{M}_J$, one can work only on consecutive ordered elements. Indeed, consider r consecutive elements $c^l, c^{l+1}, ..., c^{l+r-1}$, with $r \geq 2$. Suppose that for each k with $l \leq k \leq l+r-2$: $c^k \oplus c^{k+1} \in \mathcal{M}_J$. Since $\mathcal{M}_J$ is a subspace, it follows immediately that for each s, t with $l \leq s, t \leq l+r-2$ $c^s \oplus c^t \in \mathcal{M}_J$. Thus, given $r \geq 2$ consecutive elements that belong to the same coset of $\mathcal{M}_J$, it follows that $\binom{r}{2} = \frac{r \cdot (r-1)}{2}$ different pairs belong to the same coset of $\mathcal{M}_J$. In the same way, consider r consecutive elements $c^l, c^{l+1}, ..., c^{l+r-1}$ with $r \geq 2$, such that $c^k \oplus c^{k+1} \notin \mathcal{M}_J$ for each$k$ with $l \leq k \leq l+r-2$. Since $\mathcal{M}_J$ is a subspace, it follows immediately that $c^s \oplus c^t \notin \mathcal{M}_J$ for each s, t with $l \leq s, t \leq l+r-2$.

In other words, thanks to the ordering algorithm, it is possible to work only on $2^{32} - 1$ pairs (i.e. the pairs composed of two consecutive elements), but at the same time to have information on all the $2^{31} \cdot (2^{32} - 1) \simeq 2^{63}$ different pairs. The pseudo-code of such algorithm is given in Algorithm 2.

What is the total computational cost of this procedure? Given a set of n ordered elements, the computational cost to count the number of pairs that belong to the same coset of $\mathcal{M}_J$ is well approximated by n look-ups table, since one works only on consecutive elements. Using the *merge sort* algorithm to order this set (which has a computational cost of $O(n \log n)$ memory access), the total computational cost for the verifier is approximately of $n \cdot (1 + \log n)$ table look-ups. In our case, since the verifier has to consider a single coset of $\mathcal{D}_I$ of 2^{32} elements and to repeat this procedure four times (i.e. one for each $\mathcal{M}_J$ with $|J| = 3$), the cost is well approximated by $4 \cdot 2^{32} \cdot (1 + \log 2^{32}) = 2^{39}$ table look-ups, or equivalently $2^{32.4}$ five-round encryptions of AES (using the approximation 20 table look-ups $\approx$ 1 round of AES).

Practical Tests on small-scale AES

All the attacks that we are going to present have been practically tested on small-scale AES[20]. The practical results are in accordance with the theoretical ones.

5.9.2. Multiple-of-n Key-Recovery Attack

Consider 2^{16} plaintexts in the same coset of $\mathcal{D}_I \cap \mathcal{C}_j$ for $|j| = 1$ and $|I| = 2$ - e.g. $\mathcal{D}_{0,1} \cap \mathcal{C}_0$, and the corresponding ciphertexts after 4-round. As proved in Sect. 5.1, the number of different pairs of ciphertexts that belong to the same coset of $\mathcal{M}_K$ for $|K| = 3$ is always a multiple of 2 (or 4 if $|I| = 3$), while it can take any possible value for a random permutation.

The idea of the attack is to guess 4 bytes of the key (i.e. the j-th column), to partially decrypt $\mathcal{D}_I \cap \mathcal{C}_j$ and to ask for the corresponding ciphertexts. Since for a wrong key, the behavior is similar to the one of a random permutation - the number of collisions is not a multiple of 2 with prob. 1, it is possible to filter wrong keys and to find the right one.

[20] The source codes of the attacks are available at `https://github.com/Krypto-iaik/Distinguisher_5RoundAES`

```
Data: $2^{32}$ (plaintext, ciphertext) pairs $(p^i, c^i)$ for $i = 0, ..., 2^{32} - 1$ in a single coset of $\mathcal{D}_I$
      with $|I| = 1$.
Result: Number of collisions $n$
for all $J$ with $|J| = 3$ do
    Re-order the $2^{32}$ (plaintexts, ciphertexts) pairs using the partial order relationship $\preceq$
      defined in Def. 14;                                        // $\preceq$ depends on $J$
    Let $(\tilde{p}^i, \tilde{c}^i)$ for $i = 0, ..., 2^{32} - 1$ the order (plaintext, ciphertext) pairs;
    $n \leftarrow 0$;                        // $n$ denotes the number of collisions in $\mathcal{M}_J$
    $i \leftarrow 0$;
    while $i < 2^{32}$ do
        $r \leftarrow 1$;
        $j \leftarrow i$;
        while $\tilde{c}^j \oplus \tilde{c}^{j+1} \in \mathcal{M}_J$ do
            $r \leftarrow r + 1$;
            $j \leftarrow j + 1$;
        end
        $i \leftarrow j + 1$;
        $n \leftarrow n + r \cdot (r - 1)/2$;
    end
end
return $n$.
```

Algorithm 2: Count the number of collisions *by re-ordering the pairs of texts.*

Data Cost. Given a single coset of $\mathcal{D}_I \cap \mathcal{C}_j$, the probability that the number of collisions is a multiple of 2 is 1/2 for a wrong key. Thus, the probability that a wrong key survives n tests is 2^{-n}. Since there are 2^{32} different keys to test, $n \geq 32$ tests are sufficient to filter all the wrong keys with good probability. Since each coset of $\mathcal{C}_j$ contains 2^{16} different cosets of $\mathcal{D}_I \cap \mathcal{C}_j$, it follows that 2^{32} chosen plaintexts in the same coset of $\mathcal{D}_j$ are sufficient to find one diagonal (remember that each coset of $\mathcal{D}_j$ is mapped into a coset of $\mathcal{C}_j$ after one round). Using this strategy to find three diagonals of the key (one diagonal is found by brute force), the data complexity is of $2^{33.6}$ chosen plaintexts.

Computational Cost. Using Algorithm 2, the computational cost of the attack is well approximated by the cost of the re-ordering step for each possible key. In particular, in order to find one diagonal of the key, the cost can be approximated by $2^{32}\cdot 2^{16}\cdot(2+\log 2^{16})\cdot(1+1/2+1/4+1/8+...) \simeq 2^{53.1}$ table look-ups. Thus, the total cost is $3 \cdot 2^{53.1} \cdot (5 \cdot 20)^{-1} + 2^{32} \simeq 2^{48}$ five-round encryption to find the entire key (by assuming 20 table look-ups $\approx$ 1 encryption). The term $1 + 1/2 + 1/4 + 1/8 + ...$ is due to the fact that after the 1st test only 1/2 of the possible keys survived, after the 2nd test only 1/4 of the possible keys survived and so on. Indeed, note that the number of collisions is a multiple of 2 only with probability 1/2. In other words, after the 1st test one repeats the process for $2^{32}/2 \simeq 2^{31}$ keys, after the 2nd test one repeats the process for $2^{32}/4 \simeq 2^{30}$ keys and so on. This result has been checked also by practical tests.

5.9.3. Truncated Diff. Attack based on the *Mean*

Here we exploit the fact the average number of collisions is (a little) bigger for the right key than for a wrong guessed key, i.e. *we propose the first truncated differential attack on 5-round AES (that exploits a differential trail with probability different from zero).*

Consider 2^{24} plaintexts in the same coset of $\mathcal{D}_I \cap \mathcal{C}_j$ for $|j| = 1$ and $|I| = 3$ - e.g. $\mathcal{D}_{0,1,2} \cap \mathcal{C}_0$, and the corresponding ciphertexts after 4-round. As we have just seen in Sect. 5.3, the average number of different pairs of ciphertexts that belong to the same coset of $\mathcal{M}_K$ for $|K| = 3$ is approximately 32 770.524 versus 32 767.998 in the random case. In other words, the probability that a pair of ciphertexts belongs to the same coset of $\mathcal{M}_K$ for $|K| = 3$ is $2^{-32} + 2^{-45.6625}$ for AES versus 2^{-32} for the random case/wrong guessed key.

The idea of the attack is to guess 4 bytes of the key (i.e. the j-th diagonal), to partially decrypt $\mathcal{D}_I \cap \mathcal{C}_j$ and to ask for the corresponding ciphertexts. Exploiting the previous property that holds on the ciphertexts, it is possible to filter wrong keys and to find the right one. We expect that the number of collisions is bigger for the right key of AES than for a wrong one. Indeed, if the key is wrong, then the texts are distributed in several cosets of $\mathcal{D}_I \cap \mathcal{C}_j$ after one round (not in only one), and one gets the same behavior that occurs for a random permutation. In particular, we emphasize that our truncated differential distinguisher proposed in this paper works if and only if one consider an entire initial coset of $\mathcal{D}_I \cap \mathcal{C}_j$.

Data Cost. Assume that the goal is to find the right key with probability bigger than 95%[21], and assume that the behavior for a wrong guessed key is the same of a random permutation. Since one works on 4 bytes of the key, one has to use the secret-key distinguisher $4 \cdot 2^{32} = 2^{34}$ different times. In other words, the data cost is approximately given by formula (6.9) where $prob = 0.95^{1/2^{34}}$. It follows that for $p_{rand} \simeq 2^{-30} - 3 \cdot 2^{-63}$ and $p_{AES} \simeq 2^{-30} + 2^{-43.6625}$, the number of different pairs that one needs to use in order to set up the attack is $n \geq 3 \cdot 2^{59.43}$ (where the factor 3 is due to the observations given in Sect. 5.7.3). Since there are 4 different subspace $\mathcal{D}_I \cap \mathcal{C}_j$ and since each coset of $\mathcal{D}_I \cap \mathcal{C}_j$ contains approximately $\binom{2^{24}}{2} \simeq 2^{47}$ different pairs after one round, one needs approximately $2^{12.02}$ different initial cosets or approximately $2^{34.02}$ chosen plaintexts in the same coset of $\mathcal{D}_j$ in order to find one diagonal of the key. If two diagonals are found by brute force, the cost of finding the entire key is of $2 \cdot 2^{34.02} = 2^{35}$ chosen plaintexts.

Computational Cost. Using Algorithm 2, the computational cost of the attack is well approximated by the cost of the re-ordering step for each possible key. In particular, in order to find one diagonal, the cost can be approximated by $4 \cdot 2^{12.02} \cdot 2^{32} \cdot 2^{24} \cdot (2 + \log 2^{24}) \simeq 2^{74.7}$ table look-ups. Thus the total cost is $2 \cdot 2^{74.7} \cdot (5 \cdot 20)^{-1} + 2^{64} \simeq 2^{69.2}$ five-round encryption to find the entire key (by assuming 20 table look-ups $\approx$ 1 encryption).

5.9.4. Truncated Diff. Attack based on the *Variance*

Here we exploit the fact the variance is higher for the right key than for a wrong guessed key.

Consider 2^{24} plaintexts in the same coset of $\mathcal{D}_I \cap \mathcal{C}_j$ for $|j| = 1$ and $|I| = 3$ - e.g. $\mathcal{D}_{0,1,2} \cap \mathcal{C}_0$, and the corresponding ciphertexts after 4-round. What is the variance of the number of collisions in the same coset of $\mathcal{M}_K$ for $|K| = 3$ after 4 rounds? To compute a good approximation of the variance, we re-use the same calculation proposed in Sect. 5.4.1. For this reason, we refer to that section for all the details and we give here only the final result.

Assume K fixed. For a wrong guessed key, the variance is well approximated by

$$Var_{wrongKey} = 2^{23} \cdot (2^{24} - 1) \cdot 2^{-32} \cdot (1 - 2^{-32}) \simeq 2^{15},$$

that is the standard deviation is equal to $\delta_{wrongKey} = 2^{7.5}$. What about right key guessed? Given 2^{24} plaintexts, there are $3 \cdot 2^{23} \cdot (2^8 - 1)^2 = 2^{40.58}$ different pairs with one equal generating variable and $2^{23} \cdot (2^8 - 1)^3 = 2^{46.99}$ different pairs with different generating variables. The variance is given by

$$\begin{aligned} Var_{rightKey} = {} & 4^2 \cdot 2^{44.99} \cdot (2^{-32} - 2^{-45.6625}) \cdot (1 - 2^{-32} + 2^{-45.6625}) + \\ & + (2^9)^2 \cdot 2^{30.58} \cdot (2^{-32} - 2^{-45.6625}) \cdot (1 - 2^{-32} + 2^{-45.6625}) \simeq 2^{17.8}, \end{aligned}$$

that is the standard deviation is equal to $\delta_{rightKey} = 2^{8.9}$. This difference can be exploited to find the right key. In order to derive concrete number for data and computational complexity, as for the secrete-key distinguisher, we consider the results on small-scale AES.

[21] In other words, we assume that the maximum number of collisions occurs for the right key with probability 95%.

For small-scale AES - denoted in the following by symbol *, consider as before 2^{12} plaintexts in the same coset of $\mathcal{D}_I \cap \mathcal{C}_j$ for $|j| = 1$ and $|I| = 3$ and assume K fixed. For a wrong guessed key, the variance is well approximated by

$$Var^\star_{wrongKey} = 2^{11} \cdot (2^{12} - 1) \cdot 2^{-16} \cdot (1 - 2^{-16}) \simeq 2^7,$$

that is the standard deviation is equal to $\delta^\star_{wrongKey} = 2^{3.5}$. What about right key guessed? Given 2^{12} plaintexts, there are $3 \cdot 2^{11} \cdot (2^4 - 1)^2 = 2^{20.4}$ different pairs with one equal generating variable and $2^{11} \cdot (2^4 - 1)^3 = 2^{22.7}$ different pairs with different generating variables. The variance is given by

$$\begin{aligned} Var^\star_{rightKey} = {} & 4^2 \cdot 2^{20.7} \cdot (2^{-16} - 2^{-24.67}) \cdot (1 - 2^{-16} + 2^{-24.67}) + \\ & + (2^5)^2 \cdot 2^{15.4} \cdot (2^{-16} - 2^{-24.67}) \cdot (1 - 2^{-16} + 2^{-24.67}) \simeq 2^{10.1}, \end{aligned}$$

that is the standard deviation is equal to $\delta^\star_{rightKey} = 2^{5.05}$.

Data and Computational Costs. As for the secret-key distinguisher of Sect. 5.7.1, the ratio between the standard deviation is similar for the small scale AES and full-size AES

$$\frac{2^{8.9}}{2^{7.5}} \approx 2.75 \approx \frac{2^{5.05}}{2^{3.5}}.$$

Thus, we use our results on small-scale AES to derive concrete numbers for the full-size AES case. By practical tests, we have found that $\geq 2^6$ initial cosets are sufficient to have a good estimation of the variance/standard deviation. Since for each initial coset it is possible to compute the number of collisions in $\mathcal{M}_J$ for each J with $|J| = 3$, at least 2^6 initial cosets are largely sufficient to set up the distinguisher. Due to the relation between small-scale AES and full-size AES previously discussed, we claim that the data cost to distinguish to find one diagonal of the key is of 2^{32} chosen plaintexts in the same coset of $\mathcal{D}_j$ (observe that after one round, it contains $4 \cdot 2^8$ different cosets of $\mathcal{D}_I \cap \mathcal{C}_j$). If two diagonals are found by brute force, the total data cost is well approximated by 2^{33} chosen plaintexts.

The computational cost is well approximated by the cost to compute the number of collisions for each possible key. Using Algorithm 2, the cost of finding one diagonal is well approximated by $2^{32} \cdot 2^6 \cdot 2^{24} \cdot (2 + \log 2^{24}) \simeq 2^{66.7}$ table look-ups, that is the total cost is well approximated by $2 \cdot 2^{66.7} \cdot (100)^{-1} + 2^{64} \simeq 2^{64.2}$ five-round encryption to find the entire key by assuming 20 table look-ups $\approx$ 1 encryption.

6 Mixture Differential Cryptanalysis

"Multiple-of-8" distinguisher [GRR17] proposed at Eurocrypt 2017 by Grassi, Rechberger and Rønjom is the first 5-round secret-key distinguisher in the literature for AES that exploits a property which is independent of the secret key and of the details of the S-Box. As shown in the previous section, this distinguisher is based on a new structural property for up to 5 rounds of AES: by appropriate choices of a number of input pairs it is possible to make sure that the number of times that the difference of the resulting output pairs lie in a particular subspace is *always* a multiple of 8. On the other hand, as this distinguisher is based on a property that involves the whole state both in the input and in the output of AES, it makes it challenging to convert it into a key-recovery attack over more rounds, since e.g. it requires guessing the whole subkey in the initial/last round.

In [Gra17b; Gra18b] we introduced "*mixture differential cryptanalysis*" on round-reduced AES-like ciphers, a way to translate the (complex) "multiple-of-8" 5-round distinguisher [GRR17] into a simpler and more convenient one (though, on a smaller number of rounds). As we are going to show, such new proposed technique leads to a new distinguisher and key-recovery attacks on 4- and 5-round AES (respectively) with data and computational complexity similar than other attacks in literature.

Why is it (rather) *hard* to set up key-recovery attacks that exploit such distinguisher? Given the 5-round multiplie-of-8 distinguisher, a natural question regards the possibility to exploit it in order to set up a key-recovery attack on 6-round AES-128 better than a brute force one. A possible way is the following. Consider 2^{32} chosen plaintexts in the same coset of a diagonal space $\mathcal{D}_i$, and the corresponding ciphertexts after 6 rounds. A possibility is to guess the final key, decrypt the ciphertexts and check if the number of collisions in the same coset of $\mathcal{M}_J$ is a multiple of 8. If not, the guessed key is wrong. However, since a coset of $\mathcal{M}_J$ is mapped into the full space, it seems hard to check this property one round before without guessing the entire key. It follows that it is rather hard to set up an attack different than a brute force one that exploits directly the 5-round distinguisher proposed in [GRR17]. For comparison, note that such a problem does not arise for the other distinguishers for up to 4-round AES (e.g. the impossible differential or the integral ones) present in the literature, for which it is sufficient to guess only part of the secret key in order to verify if the required property is satisfied or not.

6.1. Preview

Before going into the details, we briefly introduce the concept behind "mixture differential cryptanalysis".

6.1.1. Mixture Differential Cryptanalysis

Consider 4-round AES. Mixture differential distinguishers work as follows. Given plaintexts in the same coset of a subspace $\mathcal{C}$, the attacker first constructs all possible pairs of two (plaintexts, ciphertexts) and divides them into sets of $N \geq 2$ *non-independent* pairs. These sets are defined such that particular relationships (that involve differential and linear relationships) hold among the plaintexts of the pairs that belong to the same set. Due to the particular way - explained in detail

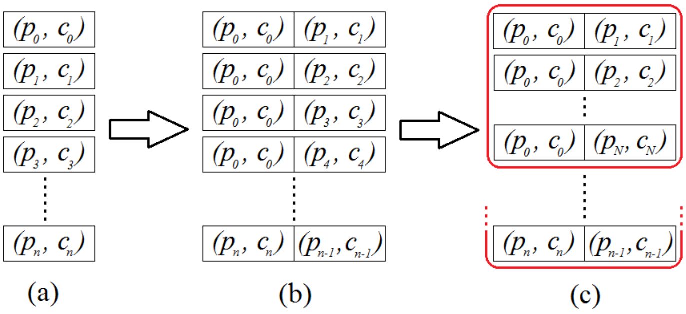

(p_0, c_0)
(p_1, c_1)
(p_2, c_2)
(p_3, c_3)
(p_n, c_n)
(a)
(p_0, c_0) (p_1, c_1)
(p_0, c_0) (p_2, c_2)
(p_0, c_0) (p_3, c_3)
(p_0, c_0) (p_4, c_4)
(p_n, c_n) (p_{n-1}, c_{n-1})
(b)
(p_0, c_0) (p_1, c_1)
(p_0, c_0) (p_2, c_2)
(p_0, c_0) (p_N, c_N)
(p_n, c_n) (p_{n-1}, c_{n-1})
(c)

Figure 6.1.: *New Differential Secret-Key Distinguishers for round-reduced AES.* Consider n (plaintexts, ciphertexts) (a). In a "classical" differential attack (b), one works independently on each pair of two (plaintexts, ciphertexts), and exploits the probability that it satisfies a certain differential trail. In our attack (c), one divides the pairs into non-random sets, and exploits particular relationships (based on differential trails) that hold among the pairs that belong to the same set in order to set up a distinguisher.

the following - in which these sets are defined, we call our new technique as "mixture differential cryptanalysis". As already pointed out, the way in which these sets are constructed resemble the "multiple-of-8" distinguisher [GRR17] recently proposed at Eurocrypt 2017.

Such sets have the property that the two ciphertexts of a certain pair belong to the same coset of a particular subspace $\mathcal{M}$ if and only if the two ciphertexts of all the other pairs in that set have the same property. In other words, given a set of pairs, it is not possible that two ciphertexts of some pairs belong to the same coset of $\mathcal{M}$, and that two ciphertexts of other pairs do not have this property. Since this last event can occur for a random permutation, it is possible to distinguish 4-round AES from a random permutation.

In more detail and referring to Fig. 6.1, given n chosen (plaintexts, ciphertexts), in a "classical" (differential) attack one works on each pair of two (plaintext, ciphertext) independently of the others - case (b). In our distinguishers/attacks instead, one first divides the pairs in (non-random) sets of $N \geq 2$ pairs of texts - case (c), and then she works on each set of pairs independently of the other sets, exploiting the property just given.

Relation with other Attacks/Distinguishers in the Literature

To the best of our knowledge, the concept of mixture differential cryptanalysis is new and has not been used in cryptanalysis before. Nonetheless there are other works that share some similarities with mixture differential cryptanalysis.

Differential Attacks. Differential attacks [BS90] exploit the fact that pairs of plaintexts with certain differences yield other differences in the corresponding ciphertexts with a non-uniform probability distribution. The resulting pair of differences is called a *differential*. Such a property can be used both to distinguish a cipher permutation from a random one, and to recover the secret key. Possible variants of this attack/distinguisher are the truncated differential attack [Knu94], in which the attacker considers only part of the difference between pairs of texts (i.e. a differential attack where only part of the difference in the ciphertexts can be predicted), and impossible differential attack [Knu98; BBD+98], in which the attacker considers differential with zero-probability.

In the original version of differential cryptanalysis [BS90], a unique differential is exploited. A generalization of such attack is multiple differential cryptanalysis [BG11], where several input differences are considered together and the corresponding output differences can be different from an input difference to another, that is the set of considered differentials has no particular structure.

The common feature of all these distinguishers/attacks is the fact that - in all these cases - the attacker focuses on the probability that a single pair of plaintexts with a certain input difference yield other difference in the corresponding pair of ciphertexts, working *independently* on each pair of texts.

Recent Results. Recently, new differential distinguishers have been proposed in the literature, precisely the polytopic cryptanalysis [Tie16a] at Eurocrypt 2016 and the yoyo distinguisher on SPN constructions [RBH17] at Asiacrypt 2017, which present an important difference with respect to the previously recalled attacks. Instead of working on each pair of two (plaintexts, ciphertexts) independently of the others as in the previous scenario, in these cases the attacker works on the relations that hold among the pairs of texts. In other words, *given a pair of two (plaintexts, ciphertexts) with a certain input/output differences, one focuses and studies how such pair influences other pairs of texts to satisfy particular input/output differences.*

More precisely, polytopic cryptanalysis is similar to multiple differential cryptanalysis. However, as opposed to assuming independence of the differentials (which does not hold in general, as shown in [Mur11]), the authors explicitly take their correlation into account and use it in their framework, considering interdependencies between larger sets of texts and as they traverse through the cipher.

The strategy exploited by the yoyo game on SPN constructions proposed at Asiacrypt 2017 is similar to the one that we are going to exploit to set up our new distinguisher. Given a pair of chosen plaintexts and the corresponding ciphertexts, the attacker constructs new pair of ciphertexts related to the other ones by linear and differential relations. Authors prove that the corresponding new pair of plaintexts of this new second pair of ciphertexts satisfies - with prob. 1 - a difference related "in some sense" to the input difference of the original pair of plaintexts, independently of the secret-key. This allows to distinguish e.g. round-reduced AES from a random permutation, or to set up key-recovery attacks.

As a result, "mixture differential cryptanalysis" is similar in nature to polytopic cryptanalysis and the yoyo distinguishers. More details are given in the following.

6.1.2. Probabilistic Mixture Differential Cryptanalysis

Using the 4-round distinguisher just (roughly) presented as starting point, in [Gra17b] we presented three different properties that can be exploited to distinguish 5-round AES from a random permutation. As before, given sets of $N \geq 2$ *non-independent* pairs of two (plaintexts, ciphertexts), it is possible to prove the following[1]:

Probabilistic Mixture Differential: consider the number of sets for which two ciphertexts of at least one pair belong to the same coset of particular subspace $\mathcal{M}$; if the sets are properly defined, then this number of sets is (a little) lower for 5-round AES than for a random permutation (details are given in Sect. 6.4);

Threshold Mixture Differential [Gra17b]: consider the number of sets with the following property: the number of pairs for which the two ciphertexts belong to the same coset of a particular subspace $\mathcal{M}$ is higher than a certain threshold $Z \in \mathbb{N}$; if this number Z and the sets are properly defined, then this number of sets is higher for 5-round AES than for a random permutation;

Impossible Mixture Differential [Gra17b]: if the sets are properly defined, for 5-round AES there exists at least one set for which the two ciphertexts of *all* pair in that set do not belong to the same coset of a particular subspace $\mathcal{M}$; in contrast, for a random permutation, for each

[1] In this thesis, we limit ourselves to present *only* the "Probabilistic Mixture Differential" distinguisher for 5-round AES, since it is the only distinguisher that can be used to set up a key-recovery attack on 6-round AES-128 faster than brute force. The details of the other two distinguishers can be found in [Gra17b].

set there exists *at least* one pair for which the two ciphertexts belong to the same coset of a particular subspace $\mathcal{M}$.

Even if such 5-round distinguishers have higher complexity than e.g. the "multiple-of-8" one, the first one can be used as *starting point to set up the first key-recovery attack on 6-round AES that exploits directly a 5-round secret-key distinguisher* (which is independent of the secret key).

6.1.3. Key-Recovery Attacks

Finally, mixture differential cryptanalysis is not only theoretically intriguing, but indeed relevant for practical cryptanalysis. In particular, new (competitive) key-recovery attacks can be set up using (probabilistic) mixture differential distinguishers. In this attack, the attacker chooses plaintexts in the same coset of a particular subspace $\mathcal{D}$ which is mapped after one round into a coset of another subspace $\mathcal{C}$. Using the mixture differential distinguisher just introduced and the facts that

- the way in which the pairs are divided in sets depends on the (partially) guessed key
- the behavior of a set for a wrongly guessed key is (approximately) the same as the case of a random permutation,

she can filter wrong candidates of the key, and finally finds the right one.

This attack on 5-round AES has then been improved in [BDK+18], becoming the one with the *lowest computational cost* among the attacks currently present in the literature (that do not use adaptive chosen plaintexts/ciphertexts). More details are given in the following.

6.2. New 4-round Secret-Key Distinguisher for AES

First of all, we re-exploit the multiple-of-8 property proposed in [GRR17] to set up a *new* 4-round secret-key distinguisher for AES. Before we go into the details, we present the general idea.

As we have just seen, given 2^{32} plaintexts in the same coset of $\mathcal{M}_I$ for $|I| = 1$ and the corresponding ciphertexts after 1 round, that is (p^i, c^i) for $i = 0, ..., 2^{32} - 1$ where $p^i \in \mathcal{M}_I \oplus a$ and $c^i = R(p^i)$, then the number n of different pairs of ciphertexts (c^i, c^j) for $i \neq j$ that satisfy $c^i \oplus c^j \in \mathcal{D}_J$ is always a multiple of 8. This is due to the fact that if one pair of texts belong to the same coset of $\mathcal{D}_J$ after one round, then other pairs of texts have the same property.

Thus, consider a pair of plaintexts p^1 and p^2 such that the corresponding texts after one round belong (or not) to the same coset of $\mathcal{D}_J$. As we have seen, there exist other pairs of plaintexts $\hat{p}^1$ and $\hat{p}^2$ whose ciphertexts after one round have the same property. *The crucial point is that the pairs* (p^1, p^2) *and* $(\hat{p}^1, \hat{p}^2)$ *are not independent in the sense that the variables that generate the first pair of texts are the same that generate the other pairs, but in a different combination.* The idea is to exploit this property in order to set up a new distinguisher for round-reduced AES. In other words, *instead of just counting the number of collisions and check that it is a multiple of* 8 *as in [GRR17], the idea is to check if these relationships between the variables that generate the plaintexts* (whose ciphertexts belong or not the same coset of a given subspace $\mathcal{M}_J$) *hold or not.*

6.2.1. *Mixture Differential* Distinguisher for 4-round AES

A formal description of the proposed Mixture Differential Distinguisher for 4-round AES is given in the following Theorem.

Theorem 7 ([Gra17b; Gra18b]). *Given the subspace $\mathcal{C}_0 \cap \mathcal{D}_{0,3} \equiv \langle e_{0,0}, e_{1,0} \rangle \subseteq \mathcal{C}_0$, consider two plaintexts p^1 and p^2 in the same coset $(\mathcal{C}_0 \cap \mathcal{D}_{0,3}) \oplus a$ generated by $p^1 \equiv (z^1, w^1)$ and $p^2 \equiv (z^2, w^2)$. Let $\tilde{p}^1, \tilde{p}^2 \in \mathcal{C}_0 \oplus a$ be two other plaintexts generated by*

$$\tilde{p}^1 \equiv (z^1, w^1, x, y),\ \tilde{p}^2 \equiv (z^2, w^2, x, y) \quad or \quad \tilde{p}^1 \equiv (z^1, w^2, x, y),\ \tilde{p}^2 \equiv (z^1, w^2, x, y)$$

where x and y can take any possible value in $\mathbb{F}_{2^8}$. The following event

$$R^4(p^1) \oplus R^4(p^2) \in \mathcal{M}_J \quad \textit{if and only if} \quad R^4(\tilde{p}^1) \oplus R^4(\tilde{p}^2) \in \mathcal{M}_J$$

holds with prob. 1 for 4-round AES, independently of the secret key, of the details of the S-Box and of the MixColumns matrix (except for the branch number equal to 5).

Since for a random permutation the same event happens with approximately probability $2^{-32\cdot(4-|J|)}$ - i.e close to 0 (note that this probability is maximized by $|J| = 3$), it is possible to exploit this fact to set up a 4-round distinguisher. Due to the fact that the variables of p^1 and p^2 are "mixed" in order to generate $\hat{p}^1$ and $\hat{p}^2$, we name this distinguisher as *Mixture Differential* distinguisher.

Moreover, it is also possible to provide similar theorems for the case in which no generating variables are equal or a single generating variable is equal.

Theorem 8 ([Gra17b; Gra18b]). *Given the subspace $\mathcal{C}_0 \equiv \langle e_{0,0}, e_{1,0}, e_{2,0}, e_{3,0}\rangle$, consider two plaintexts p^1 and p^2 in the same coset $\mathcal{C}_0 \oplus a$ generated by $p^1 \equiv (x^1, y^1, z^1, w^1)$ and $p^2 \equiv (x^2, y^2, z^2, w^2)$. Let $\tilde{p}^1, \tilde{p}^2 \in \mathcal{C}_0 \oplus a$ be two other plaintexts generated by*

1. (x^2, y^1, z^1, w^1) *and* (x^1, y^2, z^2, w^2);
2. (x^1, y^2, z^1, w^1) *and* (x^2, y^1, z^2, w^2);
3. (x^1, y^1, z^2, w^1) *and* (x^2, y^2, z^1, w^2);
4. (x^1, y^1, z^1, w^2) *and* (x^2, y^2, z^2, w^1);
5. (x^2, y^2, z^1, w^1) *and* (x^1, y^1, z^2, w^2);
6. (x^2, y^1, z^2, w^1) *and* (x^1, y^2, z^1, w^2);
7. (x^2, y^1, z^1, w^2) *and* (x^1, y^2, z^2, w^1).

The following event

$$R^4(p^1) \oplus R^4(p^2) \in \mathcal{M}_J \quad \textit{if and only if} \quad R^4(\tilde{p}^1) \oplus R^4(\tilde{p}^2) \in \mathcal{M}_J$$

holds with prob. 1 for 4-round AES, independently of the secret key, of the details of the S-Box and of the MixColumns matrix (except for the branch number equal to 5).

Remark. We highlight that *the proof of the previous theorems follows immediately from the proof of the multiple-of-8 property given in Sect. 5.1.1.* For completeness, we mention that a detailed proof can be found in [Gra18b, Sect. 4.1.1].

Data and Computational Cost

Data Cost. Since a coset of $\mathcal{C}_0 \cap \mathcal{D}_{0,3}$ contains 2^{16} plaintexts, it is possible to construct $2^{15}\cdot(2^{16}-1) \simeq 2^{31}$ different pairs. For our goal, we consider only the pairs of texts $p^1 \equiv (z^1, w^1)$ and $p^2 \equiv (z^2, w^2)$ with different generating variables[2], that is $z^1 \neq z^2$ and $w^1 \neq w^2$. The number of pairs with two different generating variables is approximately given by $\binom{2^{16}}{2} \approx 2^{31}$, where note that only half of them are independent.

In order to distinguish 4-round AES from a random permutation, one has to check that

$$c^1 \oplus c^2 = R^4\big(p^1 \equiv (z^1, w^1)\big) \oplus R^4\big(p^2 \equiv (z^2, w^2)\big) \in \mathcal{M}_J$$

if and only if

$$\hat{c}^1 \oplus \hat{c}^2 = R^4\big(\hat{p}^1 \equiv (z^1, w^2)\big) \oplus R^4\big(\hat{p}^2 \equiv (z^2, w^1)\big) \in \mathcal{M}_J.$$

If this property is not satisfied for at least one pair, then it is possible to conclude that the analyzed permutation is a random one.

[2] If $z^1 = z^2$ or $w^1 = w^2$, then $p^1 \oplus p^2 \in \big(\mathcal{C}_0 \cap \mathcal{D}_k\big) \subseteq \mathcal{D}_k$ for a certain $k \in \{0, 3\}$, which implies that $R^4(p^1) \oplus R^4(p^2) \notin \mathcal{M}_J$ for each J due to the "impossible differential trail" given in (4.6).

```
Data: 2 cosets of $\mathcal{D}_{0,3} \cap \mathcal{C}_0$ (e.g. $(\mathcal{C}_0 \cap \mathcal{D}_{0,3}) \oplus a_i$ for $a_0, a_1 \in (\mathcal{D}_{0,3} \cap \mathcal{C}_0)^{\perp}$) and
      corresponding ciphertexts after 4 rounds
Result: $0 \equiv$ Random permutation or $1 \equiv$ 4-round AES - Prob. 95%
for each coset $(\mathcal{D}_{0,3} \cap \mathcal{C}_0) \oplus a_x$ for $x = 0, 1$ do
    for each $I \subseteq \{0, 1, 2, 3\}$ with $|I| = 3$ do
        let $(p^i, c^i)$ for $i = 0, ..., 2^{16} - 1$ be the $2^{16}$ (plaintexts, ciphertexts) of $(\mathcal{D}_{0,3} \cap \mathcal{C}_0) \oplus a_x$;
        re-order this set of elements w.r.t. the partial order $\preceq$ described in Def. 14 s.t.
         $c^k \preceq c^{k+1}$ for each $k$;                              // $\preceq$ depends on $I$
        $i \leftarrow 0$;
        while $i < 2^{16} - 1$ do
            $j \leftarrow i$;
            while $c^j \oplus c^{j+1} \in \mathcal{M}_I$ do
                $j \leftarrow j + 1$;
            end
            for each $k$ from $i$ to $j$ do
                for each $l$ from $k + 1$ to $j$ do
                    given $p^k \equiv (z^1, w^1)$ and $p^l \equiv (z^2, w^2)$, let $q^1 \equiv (z^1, w^2)$ and $q^2 \equiv (z^2, w^1)$
                     in $(\mathcal{D}_{0,3} \cap \mathcal{C}_0) \oplus a_i$;
                    if $R^4(q^1) \oplus R^4(q^2) \notin \mathcal{M}_I$// Remember that $R^4(p^k) \oplus R^4(p^l) \in \mathcal{M}_I$ then
                        return 0.                                 // Random permutation
                    end
                end
            end
            $i \leftarrow j + 1$;
        end
    end
end
return 1.                              // 4-round AES permutation - Prob. 95%
```

Algorithm 3: *Secret-Key Distinguisher for 4-round of AES.*

Given *a random permutation* $\Pi(\cdot)$*, what is the probability that* $c^1 \oplus c^2 \equiv \Pi(p^1) \oplus \Pi(p^2) \in \mathcal{M}_J$ *and* $\hat{c}^1 \oplus \hat{c}^2 \equiv \Pi(\hat{p}^1) \oplus \Pi(\hat{p}^2) \notin \mathcal{M}_J$ *- or vice-versa - for a certain* $J \subset \{0, 1, 2, 3\}$ *with* $|J| = 3$*?* Since there are 4 different indexes J with $|J| = 3$ and since $Prob(t \in \mathcal{M}_J) = 2^{-32 \cdot (4 - |J|)}$, this event happens with probability (approximately) equal to

$$2 \cdot 4 \cdot 2^{-32} \cdot (1 - 2^{-32}) \simeq 2^{-29}.$$

As a result, in order to distinguish a random permutation from 4-round AES with probability higher than pr, it is sufficient that the previous event occurs for at least one pair of two pairs of texts with probability higher than pr (in order to recognize the random permutation). It follows that one needs approximately n different *independent* pairs of texts such that $pr \geq 1 - (1 - 2^{-29})^n$, that is

$$n \geq \frac{\log(1 - pr)}{\log(1 - 2^{-29})} \approx -2^{29} \cdot \log(1 - pr).$$

For $pr = 95\%$, one needs approximately $n \geq 2^{30.583}$ different *independent* pairs of texts, that is approximately 2 different cosets $\mathcal{C}_0 \cap \mathcal{D}_{0,3}$ for a total data cost of $2^{16} \cdot 2 = 2^{17}$ chosen plaintexts.

Computational Cost. As already done before, in order to implement the distinguisher, the idea is to re-order the ciphertexts using a particular partial order $\preceq$ as defined in Def. 14, and to work in the way described in Algorithm 3.

Instead of checking the previous property for all possible pairs of texts, the idea is to check it only for the pairs of texts for which the two ciphertexts belong to the same coset of $\mathcal{M}_J$. In other words, if $c^1 \oplus c^2 \in \mathcal{M}_J$, then one checks that $\hat{c}^1 \oplus \hat{c}^2 \in \mathcal{M}_J$ (prob. 1 for 4-round AES *vs* prob. 2^{-32} for a random permutation). Instead, if $c^1 \oplus c^2 \notin \mathcal{M}_J$, then one does not check that $\hat{c}^1 \oplus \hat{c}^2 \notin \mathcal{M}_J$. Note that the probability of this last event is very close for the AES and for the random permutation (prob. 1 for 4-round AES *vs* prob. $1 - 2^{-32}$ for a random permutation). In other words, checking that "if $c^1 \oplus c^2 \in \mathcal{M}_J$ then $\hat{c}^1 \oplus \hat{c}^2 \in \mathcal{M}_J$" is sufficient to distinguish 4-round AES from a random permutation.

The reason of this strategy - already proposed in the previous sections - is that it allows to save and minimize the computational cost, which is well approximated by $2^{23.09}$ table look-ups, or approximately $2^{16.75}$ four-round encryptions (assuming 20 table look-ups $\approx$ 1 round of encryption), where we limit ourselves to remember that the cost of sorting a set of n texts w.r.t. a given partial order is $\mathcal{O}(n \cdot \log n)$ table look-ups.

Practical Verification

Using a C/C++ implementation[3], we have practically verified the distinguishers just described both for full size AES and a small scale variant of AES, as presented in [CMR05]. While for full size AES each word is composed of 8 bits, in the small scale variant each word is composed of 4 bits (we refer to [CMR05] for a complete description of this small scale AES). We highlight that the previous results hold exactly in the same way also for this small scale variant of AES, since the previous argumentation is independent of the fact that each word of AES is of 4 or 8 bits.

The distinguisher just presented works in the same way for full and small scale AES, and it is able to distinguish AES from a random permutation using $2 \cdot (2^8)^2 = 2^{17}$ chosen plaintexts in the first case and $2 \cdot (2^4)^2 = 2^9$ in the second one (i.e. 2 cosets of $\mathcal{C}_0 \cap \mathcal{D}_{0,3}$, each one of size 2^{16} and 2^8 respectively for full and small scale AES[4]) as expected. For full size AES, while the theoretical computational cost is of 2^{23} table look-ups, the practical one is on average 2^{22} in the case of a random permutation and 2^{24} in the case of an AES permutation. We emphasize that for a random permutation, it is sufficient to find *one* pair of two pairs of texts that does not satisfy the required property (to recognize the random permutation). In the case of the AES permutation, the difference between the theoretical and the practical cases (i.e. a factor 2) is due to the fact that the cost of the merge sort algorithm is $O(n \cdot \log n)$ and by the definition of the big $O(\cdot)$ notation[5].

For the small scale AES, using 2 different initial cosets of $\mathcal{C}_0 \cap \mathcal{D}_{0,3}$, the theoretical computational cost is well approximated by $2 \cdot 4 \cdot 2^8 \cdot (\log 2^8 + 1) \simeq 2^{14.2}$ table look-ups. The practical cost is approximately $2^{13.5}$ for the case of a random permutation and 2^{15} for the AES case.

6.2.2. Comparison with Other 4-round Secret-Key Distinguishers

Here we highlight the major differences with respect to the other 4-round AES secret-key distinguishers present in the literature. Omitting the integral one (which exploits a completely different property), we focus on the impossible and the truncated differential distinguishers, on the polytopic cryptanalysis, on the "multiple-of-8" distinguisher (adapted - in a natural way - to the 4-round case) and on the yoyo distinguisher.

[3] The source code of the distinguisher is available at `https://github.com/Krypto-iaik/Attacks_AES`

[4] Following the same analysis proposed in Sect. 6.2.1, here we show that 2 initial cosets are necessary to set up the attack also for the small scale case. Similar to before, the probability that $R^4(p^1) \oplus R^4(p^2) \in \mathcal{M}_J$ and $R^4(\hat{p}^1) \oplus R^4(\hat{p}^2) \notin \mathcal{M}_J$ (or vice-versa) for a (small scale) random permutation is $2 \cdot 4 \cdot 2^{-16} \cdot (1 - 2^{-16}) = 2^{-13}$. It follows that one needs $n \geq 2^{14.583}$ different *independent* pairs of texts to set up the attack with probability higher than 95%, that is approximately 2 different cosets $\mathcal{C}_0 \cap \mathcal{D}_{0,3}$ (note that for each coset it is possible to construct $\frac{1}{2} \cdot \binom{2^8}{2} \approx 2^{14}$ independent pairs of texts).

[5] A similar difference among the theoretical and the practical cases was present also in [GRR17].

Impossible Differential. The *impossible differential distinguisher* exploits the property that $\mathcal{M}_I \cap \mathcal{D}_J = \{0\}$ for $|I| + |J| \leq 4$ (see (4.6) for details). In our case, we consider plaintexts in the same coset of $\mathcal{C}_0 \cap \mathcal{D}_I \subseteq \mathcal{D}_I$ where $|I| \geq 2$ (e.g. $I = \{0, 3\}$) and looks for collisions in $\mathcal{M}_J$ with $|J| = 3$. Since $|I| + |J| \geq 5$, the property exploited by the impossible differential distinguisher cannot be applied here.

Truncated Differential. The *truncated differential distinguisher* has instead some aspects in common with our distinguisher. In this case, given pairs of plaintexts with certain difference on certain bytes (i.e. that belong to the same coset of a subspace $\mathcal{X}$), one considers the probability that the corresponding ciphertexts belong to the same coset of a subspace $\mathcal{Y}$. For 2-round AES it is possible to exploit truncated differential trails with probability 1, while for up to 5-round there exist truncated differential trails with probability lower than 1 but higher than for the random case (in both cases, $\mathcal{X} \equiv \mathcal{D}_I$ and $\mathcal{Y} \equiv \mathcal{M}_J$).

Our distinguisher works in a similar way and exploits a similar property. However, instead of working with a single pair of texts independently of the others, in our distinguisher one basically considers sets of 2 "non-independent" pairs of texts and exploits the relationships that hold among the pairs of texts that belong to the same set.

Polytopic Cryptanalysis. *Polytopic cryptanalysis* [Tie16a] has been introduced by Tiessen at Eurocrypt 2016, and it can be viewed as a generalization of standard differential cryptanalysis. Consider a set of $d \geq 2$ pairs of plaintexts $(p^0, p^0 \oplus \alpha^1), (p^0, p^0 \oplus \alpha^2), ...(p^0, p^0 \oplus \alpha^d)$ with one plaintext in common (namely p^0), called d-poly. The idea of polytopic cryptanalysis is to exploit the probability that the input set of differences $\boldsymbol{\alpha} \equiv (\alpha^1, \alpha^2, ..., \alpha^d)$ is mapped into an output set of differences $\boldsymbol{\beta} \equiv (\beta^1, \beta^2, ..., \beta^d)$ after r rounds. If this probability[6] - *which depends on the S-Box details* - is different from the corresponding probability in the case of a random permutation, it is possible to set up distinguishers or key-recovery attacks. Impossible polytopic cryptanalysis focuses on the case in which the probability of the previous event is zero. In [Tie16a], an impossible 8-polytopic is proposed for 2-round AES, which allows to set up key-recovery attacks on 4- and 5-round AES.

Our proposed distinguisher works in a similar way, since also in our case we consider sets of "non-independent" pairs of texts and we focus on the input/output differences. However, instead of working with a set of pairs of plaintexts with one plaintext in common, we consider sets of pairs of texts for which particular relationships between the generating variables of the texts hold. Moreover, instead of considering the probability that "generic" input differences $\boldsymbol{\alpha}$ are mapped into output differences $\boldsymbol{\beta}$, the way in which the texts are divided in sets guarantees the two ciphertexts of *all* pairs satisfy or not an output (truncated) difference *independently of the S-Box details* (that is, it is not possible that some of them satisfy this output difference and some others not).

"Multiple-of-8" Distinguisher. The *"multiple-of-8" distinguisher* [GRR17] can be adapted to the 4-round case, e.g. considering plaintexts in the same coset of $\mathcal{C}_J$, counting the number of collisions of the ciphertexts in the same coset of $\mathcal{M}_I$ and checking if it is (or not) a multiple of 8. *Since our distinguisher exploits more information* (that is, the relationships that hold among the generating variables of the pairs of plaintexts in the same set, beside the fact that the previous number is a multiple of 8), its data and computational costs are lower than [GRR17], in particular 2^{17} chosen plaintexts/ciphertexts instead of 2^{33} and approximately 2^{23} table look-ups instead of 2^{40}.

[6] We mention that the probability of polytopic trails is usually much lower than the probability of trails in differential cryptanalysis, that is simple polytopic cryptanalysis can not in general outperform standard differential cryptanalysis - see Sect. 2 of [Tie16a] for details.

Yoyo Distinguisher. The basic idea exploited by the *yoyo distinguisher* [RBH17] proposed at Asiacrypt 2017 is similar to the one exploited by our distinguisher. Consider 4-round AES, where the initial and the final ShiftRows and the final MixColumns operations are omitted[7]. Given a pair of plaintexts in the same coset of a column space $\mathcal{C}_I$ - that is $p^1, p^2 \in \mathcal{C}_I \oplus a$, consider the corresponding ciphertexts c^1 and c^2 after 4 rounds. In the yoyo game, the idea is to construct a new pair of ciphertexts $\hat{c}^1$ and $\hat{c}^2$ by *swapping the columns* of c^1 and c^2. E.g., if $c^i \equiv (c^i_0, c^i_1, c^i_2, c^i_3)$ for $i = 1, 2$ where c^i_j denotes the j-th column of c^i, one can define the new pair of ciphertexts as $\hat{c}^1 \equiv (c^2_0, c^1_1, c^1_2, c^1_3)$ and $\hat{c}^2 \equiv (c^1_0, c^2_1, c^2_2, c^2_3)$. As proved in [RBH17], the corresponding plaintexts $\hat{p}^1 = R^{-4}(\hat{c}^1)$ and $\hat{p}^2 = R^{-4}(\hat{c}^2)$ belong to the same coset of $\mathcal{C}_I$ with prob. 1 for 4-round AES (that is, $\hat{p}^1 \oplus \hat{p}^2 \in \mathcal{C}_I$ with prob. 1), while this happens with prob. $2^{-32\cdot(4-|I|)}$ for a random permutation.

Our distinguisher and the yoyo one are very similar. Both ones exploit particular relationships that hold among the generating variables of a pair of texts and particular properties which depend on such relations to distinguish 4-round AES from a random permutation. However, we emphasize that while the yoyo distinguisher requires *adaptive* chosen ciphertexts in order to construct new pairs of texts related to the original one, in our case such new pairs of texts are constructed directly from the chosen plaintexts. In other words, ours distinguisher does not require adaptive chosen plaintexts/ciphertexts.

For completeness, we mention that the yoyo distinguisher can be set up for up to 6 rounds AES. Here we limit to recall the 5-round one, while we refer to [GRR17] for more details about the 6-round yoyo distinguisher. First of all, note that if the initial ShiftRows operation is not omitted for 4-round AES, then one considers plaintexts in the same coset of $\mathcal{D}_I$ (instead of $\mathcal{C}_I$). For 5-round AES, the idea is to consider texts in the same coset of $\mathcal{D}_J$. As we have just seen, after one round they are mapped into a coset of $\mathcal{D}_I$ with prob. $\binom{4}{|I|} \cdot (2^{-8})^{4|J|-|I|\cdot|J|}$. Then, using the 4-round yoyo distinguisher, one choose new ciphertexts by mixing generating variables. As a result, given a pair of plaintexts $p^1, p^2 \in \mathcal{D}_J \oplus a$ for $|J| = 1$, it is possible to prove that the probability[8] that $\hat{p}^1 \oplus \hat{p}^2 \in \mathcal{D}_K$ for a certain K with $|K| = 3$ is $2^{-26.2}$ *versus* 2^{-30} for a random permutation. As a result, $3 \cdot 2^{27.2} \simeq 2^{28.8}$ adaptive chosen ciphertexts are sufficient to distinguish the two cases.

6.3. New Key-Recovery Attack on 5-round AES

The previous 4-round secret-key distinguisher proposed in Theorem 8 can be used as starting point to set up a new (practically verified) key-recovery attack on 5-round AES. In this attack, the attacker chooses plaintexts in the same coset of a particular subspace $\mathcal{D}$ which is mapped after one round into a coset of another subspace $\mathcal{C}$. Using the mixture differential distinguisher just introduced and the facts that

- the way in which the pairs of two (plaintexts, ciphertexts) are divided in sets depends on the (partially) guessed key
- the behavior of a set for a wrongly guessed key is (approximately) the same as the case of a random permutation,

[7] The distinguisher works as well also in the case in which these linear operations are not omitted. We refer to [RBH17] for all the details.

[8] To compute the following probability, the idea is to consider all possible $I \subseteq \{0, 1, 2, 3\}$:

$$\underbrace{\sum_{|I|=1}^{3} \binom{4}{|I|} \cdot 2^{-32+8|I|} \cdot (1 - 2^{-32+8|I|}) \cdot}_{\text{1 round -- forward direction}} \overbrace{4 \cdot 2^{-8|I|}}^{\text{1 round -- backward direction}} \simeq 7 \cdot 2^{-29}.$$

she can filter wrong candidates of the key, and finally finds the right one.

W.l.o.g. consider two plaintexts p^1 and p^2 in the same coset of $\mathcal{D}_0$, e.g. $\mathcal{D}_0 \oplus a$ for $a \in \mathcal{D}_0^\perp$, such that $p^i = x^i \cdot e_{0,0} \oplus y^i \cdot e_{1,1} \oplus z^i \cdot e_{2,2} \oplus w^i \cdot e_{3,3} \oplus a$ or equivalently $p^i \equiv (x^i, y^i, z^i, w^i)$. By Lemma 3, there exists $b \in \mathcal{C}_0^\perp$ such that

$$R(p^i) = \begin{bmatrix} \hat{x}^i & 0 & 0 & 0 \\ \hat{y}^i & 0 & 0 & 0 \\ \hat{z}^i & 0 & 0 & 0 \\ \hat{w}^i & 0 & 0 & 0 \end{bmatrix} \oplus b \equiv MC \cdot \begin{bmatrix} \text{S-Box}(x^i \oplus k_{0,0}) & 0 & 0 & 0 \\ \text{S-Box}(y^i \oplus k_{1,1}) & 0 & 0 & 0 \\ \text{S-Box}(z^i \oplus k_{2,2}) & 0 & 0 & 0 \\ \text{S-Box}(w^i \oplus k_{3,3}) & 0 & 0 & 0 \end{bmatrix} \oplus b$$

for $i = 1, 2$, that is

$$R(p^i) \equiv (\hat{x}^i, \hat{y}^i, \hat{z}^i, \hat{w}^i) \equiv \hat{x}^i \cdot e_{0,0} \oplus \hat{y}^i \cdot e_{1,0} \oplus \hat{z}^i \cdot e_{2,0} \oplus \hat{w}^i \cdot e_{3,0} \oplus b.$$

The idea is to filter wrongly guessed keys of the first round by exploiting the previous distinguisher.

In particular, given plaintexts in the same coset of $\mathcal{D}_0$, the idea of the attack is simply to guess 4 bytes of the first diagonal of the secret key k, that is $k_{i,i}$ for each $i \in \{0, 1, 2, 3\}$, to (partially) compute $R_k(p^1)$ and $R_k(p^2)$ and to exploit the following consideration: *if the guessed key is the right one*, then

$$R^4\big[R_k(p^1)\big] \oplus R^4\big[R_k(p^2)\big] \in \mathcal{M}_J$$

if and only if there exist other pairs of texts $R_k(q^1)$ and $R_k(q^2)$ with the same property, that is

$$R^4\big[R_k(q^1)\big] \oplus R^4\big[R_k(q^2)\big] \in \mathcal{M}_J$$

where $R_k(q^1)$ and $R_k(q^2)$ are defined by a different combination of the generating variables of $R_k(p^1)$ and $R_k(p^2)$. If this property is not satisfied and due to the distinguisher just proposed, then it is possible to claim that the guessed key is a wrong candidate. As we are going to show, *this attack works because the variables that define the (other) pairs of texts $R_k(q^1)$ and $R_k(q^2)$ depend on the guessed key* (besides on the texts p^1 and p^2).

Details of the Attack

In the following we give all the details of the attack. As for the distinguisher just presented, consider a pair of texts p^1 and p^2 in the same coset of $\mathcal{D}_0$ such that

- $c^1 \oplus c^2 \equiv R^5(p^1) \oplus R^5(p^2) \in \mathcal{M}_J$ (observe that this condition is independent of the (partially) guessed key);
- $R(p^i) \equiv (\hat{x}^i, \hat{y}^i, \hat{z}^i, \hat{w}^i)$ for $i = 1, 2$ as before, s.t. $\hat{x}^1 \neq \hat{x}^2$, $\hat{y}^1 \neq \hat{y}^2$, $\hat{z}^1 \neq \hat{z}^2$ and $\hat{w}^1 \neq \hat{w}^2$.

For completeness, we emphasize that the attack works even if one or two generating variables of $R(p^1)$ and $R(p^2)$ are equal (e.g. if two generating variables are equal, in the following it is sufficient to exploit Theorem 7). We limit ourselves to discuss the case in which the generating variables are all different *only* for sake of simplicity, and since this is the event that happens with highest probability (the probability that all the generating variables are different is $[(256 \cdot 255)/256^2]^4 = \frac{255^4}{256^4} \simeq 98.45\%$). Due to the definition of $\hat{x}^i, \hat{y}^i, \hat{z}^i, \hat{w}^i$

$$[\hat{x}^i,\ \hat{y}^i\ \hat{z}^i\ \hat{w}^i]^T \equiv MC \cdot [\text{S-Box}(x^i \oplus k_{0,0}),\ \text{S-Box}(y^i \oplus k_{1,1}),\ \text{S-Box}(z^i \oplus k_{2,2}),\ \text{S-Box}(w^i \oplus k_{3,3})]^T,$$

note that the fact that “the generating variables are different” depends on the (partially) guessed key.

Given p^1 and p^2 as before, we have to define $R_k(q^1)$ and $R_k(q^2)$ in order to set up the distinguisher. Using Theorem 7 and the “super-Sbox” argumentation given in Sect. 5.1.1, it is possible to construct

7 different pairs of - intermediate - texts $R_k(q^1)$ and $R_k(q^2)$ in $\mathcal{C}_0 \oplus b$ defined by the following combinations of generating variables

1. $(\hat{x}^2, \hat{y}^1, \hat{z}^1, \hat{w}^1)$ and $(\hat{x}^1, \hat{y}^2, \hat{z}^2, \hat{w}^2)$;
2. $(\hat{x}^1, \hat{y}^2, \hat{z}^1, \hat{w}^1)$ and $(\hat{x}^2, \hat{y}^1, \hat{z}^2, \hat{w}^2)$;
3. $(\hat{x}^1, \hat{y}^1, \hat{z}^2, \hat{w}^1)$ and $(\hat{x}^2, \hat{y}^2, \hat{z}^1, \hat{w}^2)$;
4. $(\hat{x}^1, \hat{y}^1, \hat{z}^1, \hat{w}^2)$ and $(\hat{x}^2, \hat{y}^2, \hat{z}^2, \hat{w}^1)$;
5. $(\hat{x}^2, \hat{y}^2, \hat{z}^1, \hat{w}^1)$ and $(\hat{x}^1, \hat{y}^1, \hat{z}^2, \hat{w}^2)$;
6. $(\hat{x}^2, \hat{y}^1, \hat{z}^2, \hat{w}^1)$ and $(\hat{x}^1, \hat{y}^2, \hat{z}^1, \hat{w}^2)$;
7. $(\hat{x}^2, \hat{y}^1, \hat{z}^1, \hat{w}^2)$ and $(\hat{x}^1, \hat{y}^2, \hat{z}^2, \hat{w}^1)$

that must satisfy the required property

$$R^4\big[R_k(p^1)\big] \oplus R^4\big[R_k(p^2)\big] \in \mathcal{M}_J \qquad \textit{iff} \qquad R^4\big[R_k(q^1)\big] \oplus R^4\big[R_k(q^2)\big] \in \mathcal{M}_J.$$

Using this observation, it is possible to filter all the wrong keys. Again, since $R^5(p^1) \oplus R^5(p^2) \in \mathcal{M}_J$, all these pairs of - intermediate - texts $(R_k(q^1), R_k(q^2))$ must belong to the same coset of $\mathcal{M}_J$ after 4 rounds if the guessed key is the right one. If this property is not satisfied, then one can simply deduce that the guessed key is wrong (for a wrong guessed key, the behavior is similar to the one of a random permutation).

Why does the attack work? Wrong-Key Randomization Hypothesis! One of the assumption required by the proposed attack is the "wrong-key randomization hypothesis". This hypothesis states that when decrypting one or several rounds with a wrong key guess creates a function that behaves like a random function. For our setting, we formulate it as following:

Wrong-key randomization hypothesis. When the pairs of - intermediate - texts $R_k(q^1)$ and $R_k(q^2)$ are generated using a wrongly guessed key, the probability that the resulting pairs of ciphertexts satisfy the required property is equal to the probability given for the case of a random permutation.

In the following we show that such assumption holds. The crucial point is that *the new pairs of texts $R_k(q^1)$ and $R_k(q^2)$ (and the way in which they are constructed) depend on the guessed key.*

In the proposed attack, the wrong-key randomization hypothesis follows immediately from the definition of the generating variables and from the fact that the S-Box is a non-linear operation. To have more evidence of this fact, let k be the secret key and $\tilde{k}$ be a guessed key. Given $R_k(p^1) \equiv (x^1, y^1, z^1, w^1)$ and $R_k(p^2) \equiv (x^2, y^2, z^2, w^2)$ in $\mathcal{C}_0 \oplus b$ as before, the generating variables of $R_{\tilde{k}}(q^1) \equiv (\tilde{x}^1, \tilde{y}^1, \tilde{z}^1, \tilde{w}^1)$ and $R_{\tilde{k}}(q^2) \equiv (\tilde{x}^2, \tilde{y}^2, \tilde{z}^2, \tilde{w}^2)$ in $\mathcal{C}_0 \oplus b$ are given by

$$\begin{bmatrix} \tilde{x}^i \\ \tilde{y}^i \\ \tilde{z}^i \\ \tilde{w}^i \end{bmatrix} = MC \circ \text{S-Box} \circ \left(\begin{bmatrix} \tilde{k}_{0,0} \oplus k_{0,0} \\ \tilde{k}_{1,1} \oplus k_{1,1} \\ \tilde{k}_{2,2} \oplus k_{2,2} \\ \tilde{k}_{3,3} \oplus k_{3,3} \end{bmatrix} \oplus \text{S-Box}^{-1} \circ MC^{-1} \circ \begin{bmatrix} x^h \\ y^j \\ z^k \\ w^l \end{bmatrix} \right)$$

for certain $h, j, k, l \in \{1, 2\}$. For a wrongly guessed key $\tilde{k} \neq k$, the relations among the generating variables $[\tilde{x}^i, \tilde{y}^i, \tilde{z}^i, \tilde{w}^i] = [x^h, y^j, z^k, w^l]$ do *not* hold[9]. It follows that if $k \neq \tilde{k}$, then the attacker is considering *random* pairs of texts, which implies that the required property is - in general - not satisfied (as for the case of a random permutation).

Before going on, we emphasize that this result also implies the *impossibility to set up a 5-round distinguisher similar to the one just presented in this section* choosing plaintexts in the same coset of a diagonal space $\mathcal{D}_I$ instead of a column space $\mathcal{C}_I$. Indeed, given p^1 and p^2 as before in the same coset of $\mathcal{D}_I$ (instead of $\mathcal{C}_I$), since the key k is secret and the S-Box is non-linear, *there is no way to find $\hat{p}^1$ and $\hat{p}^2$ in the coset of $\mathcal{D}_I$ s.t. $R^5(p^1) \oplus R^5(p^2) \in \mathcal{M}_J$ if and only if $R^5(\hat{p}^1) \oplus R^5(\hat{p}^2) \in \mathcal{M}_J$ without guessing the secret key.*

[9] Note that if $k = \tilde{k}$, then $\tilde{x}^i = x^h$, $\tilde{y}^i = y^j$, $\tilde{z}^i = z^k$ and $\tilde{w}^i = w^l$ (which implies that the required property is satisfied) as expected.

```
Data: 1 coset of $\mathcal{D}_0$ (e.g. $\mathcal{D}_0 \oplus a$ for $a \in \mathcal{D}_0^\perp$) and corresponding ciphertexts after 5 rounds -
      more generally a coset of $\mathcal{D}_i$ for $i \in \{0,1,2,3\}$
Result: 4 bytes of the secret key - $(k_{0,0}, k_{1,1}, k_{2,2}, k_{3,3})$
let $(p^i, c^i)$ for $i = 0, ..., 2^{32} - 1$ be the $2^{32}$ (plaintexts, ciphertexts) of $\mathcal{D}_0 \oplus a$;
while more than a single candidate of the key is found - Repeat the procedure for different
 indexes $j, h$ (and $I$) // usually not necessary - only one candidate is found do
    find indexes $j$ and $h$ s.t. $c^j \oplus c^h \in \mathcal{M}_I$;
    for each one of the $2^{32}$ combinations of $\hat{k} = (k_{0,0}, k_{1,1}, k_{2,2}, k_{3,3})$ do
        (partially) compute $R_{\hat{k}}(p^j)$ and $R_{\hat{k}}(p^h)$;
        $flag \leftarrow 0$;
        for each pair $(q^1, R^5(q^1))$ and $(q^2, R^5(q^2))$ where $R_{\hat{k}}(q^1)$ and $R_{\hat{k}}(q^2)$ are constructed
         by a different combination of the generating variables of $R_{\hat{k}}(p^j)$ and $R_{\hat{k}}(p^h)$ do
            if $R^5(q^1) \oplus R^5(q^2) \notin \mathcal{M}_I$ then
                $flag \leftarrow 1$;
                next combination of $(k_{0,0}, k_{1,1}, k_{2,2}, k_{3,3})$;
            end
        end
        if $flag = 0$ then
            identify $(k_{0,0}, k_{1,1}, k_{2,2}, k_{3,3})$ as candidate of the key;
        end
    end
end
return $(k_{0,0}, k_{1,1}, k_{2,2}, k_{3,3})$
```

Algorithm 4: *5-round AES Key-Recovery Attack.* The attack exploits the 4-round "Mixture Differential" distinguisher just presented. For sake of simplicity, in this pseudo-code we limit ourselves to describe the attack of 4 bytes - 1 diagonal of the secret key (the same attack can be used to recover the entire key).

6.3.1. Data and Computational Costs

Data Cost. First of all, since the cardinality of a coset of $\mathcal{D}_I$ for $|I| = 1$ is 2^{32} and since $Prob[t \in \mathcal{M}_J] = 4 \cdot 2^{-32} = 2^{-30}$ for $|J| = 3$, the average number of collisions for each coset of $\mathcal{D}_I$ is approximately $2^{-30} \cdot \binom{2^{32}}{2} \simeq 2^{-30} \cdot 2^{63} \simeq 2^{33}$, so it's very likely that two (plaintexts, ciphertexts) (p^1, c^1) and (p^2, c^2) exist such that $c^1 \oplus c^2 \in \mathcal{M}_J$ and for which the two plaintexts have different generating variables.

Given a pair of plaintexts p^1 and p^2 for which the corresponding ciphertext c^1 and c^2 belong to the same coset of $\mathcal{M}_J$, consider the other 7 pairs of plaintexts q^1 and q^2 defined as before (that is, such that $R(q^1)$ and $R(q^2)$ are defined by a different combination of the generating variables of $R(p^1)$ and $R(p^2)$). For a wrong key, the probability that the two ciphertexts of each one of the other 7 pairs belong to the same coset of $\mathcal{M}_J$ for a fixed J (that is, the probability that a wrong key passes the test) is $(2^{-32})^7 = 2^{-224}$.

Since there are $2^{32} - 1$ wrong candidates for the diagonal of the key, the probability that at least one of them passes the test is approximately $1 - (1 - 2^{-224})^{2^{32}-1} \simeq 2^{-192}$. Thus, one pair of plaintexts p^1 and p^2 (for which the corresponding ciphertexts belong to the same coset of $\mathcal{M}_J$) together with the corresponding other 7 pairs of texts q^1 and q^2 are (largely) sufficient to discard all the wrong candidates for a diagonal of the key. Actually, in general only two different pairs q^1 and q^2 (that is, two pairs of texts given by two different combinations of the generating variables) are sufficient to discard all the wrong candidates, so it is not necessary to consider all the 7 pairs of texts q^1 and q^2. Indeed, given two pairs, the probability that at least one wrong key passes the test is approximately

$1 - (1 - 2^{-32\cdot 2})^{2^{32}-1} \simeq 2^{-32} \ll 1$, which means that all the wrong candidates are discarded with high probability.

As a result, the attack requires $2^{33.6}$ chosen plaintexts.

Computational Cost. Each coset of $\mathcal{D}_I$ with $|I| = 1$ is composed of 2^{32} texts, thus on average $2^{63} \cdot 2^{-32} = 2^{31}$ different pairs of ciphertexts belong to the same coset of $\mathcal{M}_J$ for a fixed J with $|J| = 3$. However, it is sufficient to find one collision in order to implement the attack and to find the key.

In order to find it, the best strategy is to re-order the ciphertexts with respect to the partial order $\preceq$ and then to work on consecutive elements, as done in Sect. 6.2.1. For each initial coset of $\mathcal{D}_I$ and for a fixed J, the cost of sorting the ciphertexts with respect to the partial order $\preceq$ (for $\mathcal{M}_J$ with J fixed - $|J| = 3$) and to find a collision is approximately of $2^{32} \cdot (\log 2^{32} + 1) = 2^{37}$ table look-ups.

When such a collision is found, one has to guess 4 bytes of the key and to construct - at least - two other different pairs given by a different combination of the generating variables of $R(p^1)$ and $R(p^2)$ (observe that the condition $\hat{x}^1 \neq \hat{x}^2$, $\hat{y}^1 \neq \hat{y}^2$, $\hat{z}^1 \neq \hat{z}^2$ and $\hat{w}^1 \neq \hat{w}^2$ is satisfied with probability $(255/256)^4 \approx 1$). In order to perform this step efficiently, the idea is to re-order - and to store separately *a second copy of* - the (plaintexts, ciphertexts) pairs w.r.t. the partial order $\leq$ as defined in Def. 12 s.t. $p^i \leq p^{i+1}$ for each i. Using the same strategy proposed for the 4-round distinguisher, this allows to construct these two new different pairs (and to check if the corresponding ciphertexts satisfy or not the required property) with only 4 table look-ups. As a result, the cost of this step is of $2^{32} \cdot 2 \cdot 4 = 2^{35}$ S-Box and of $2^{32} \cdot 4 = 2^{34}$ table look-ups.

It follows that the cost of finding one diagonal of the key is well approximated by 2^{35} S-Box look-ups and $2^{37.17}$ table look-ups, that is approximately $2^{30.95}$ five-round encryptions. The idea is to use this approach for three different diagonals, and to find the last one by brute force. As a result, the total computational cost is of $2^{32} + 3 \cdot 2^{30.95} = 2^{33.28}$ five-round encryptions, while the data cost is of $3 \cdot 2^{32} = 2^{33.6}$ chosen plaintexts.

Summary. As a result, the attack - practically verified on a small scale AES - requires $2^{33.6}$ chosen plaintexts and has a computational cost of $2^{33.28}$ five-round encryptions. The pseudo-code of the attack is given in Algorithm 4. We remark for completeness that the same attack works also in the decryption/reverse direction, using chosen ciphertexts instead of plaintexts.

6.3.2. Practical Verification

Using a C/C++ implementation, we have practically verified the attack just described[10] on the small scale AES [CMR05]. We emphasize that since the proposed attack is independent of the fact that each word of AES is composed of 4 or 8 bits, our verification on the small scale variant of AES is strong evidence for it to hold for the real AES.

Practical Results. For simplicity, we limit ourselves to report the result for a single diagonal of the key. First of all, a single coset of a diagonal space $\mathcal{D}_i$ is largely sufficient to find one diagonal of the key. In more detail, given two (plaintexts, ciphertexts) (p^1, c^1) and (p^2, c^2), then other two different texts q^1 and q^2 (out of the seven possible ones) are sufficient to discard all the wrong candidates of the diagonal of the key, as predicted.

About the computational cost, the theoretical cost for the small scale AES case is well approximated by $4 \cdot 2^{16} \cdot (\log 2^{16} + 1) + 2^{16} \cdot 4 = 2^{21}$ table look-ups and $2^{16} \cdot 4 \cdot 3 = 2^{19.6}$ S-Box look-ups, for a total of $2^{19.6} + 2^{21} = 2^{21.5}$ table look-ups (assuming that the cost of 1 S-Box look-up is approximately equal to the cost of 1 table look-up). The average practical computational cost is of $2^{21.5}$ table look-ups, approximately the same as the theoretical one.

[10] The source codes of the distinguishers/attacks are available at `https://github.com/Krypto-iaik/Attacks_AES`

6.3.3. Improved Key-Recovery Attack by Bar-On *et al.* (Crypto 2018)

Such attack has then been improved in [BDK+18], becoming the one with the *lowest computational cost* among the attacks currently present in the literature (that do not use adaptive chosen plaintexts/ciphertexts).

In particular, our attack just presented can break 5-round AES in data, memory and time complexities of 2^{32}. However, a variant of the Square attack [DKR97; KW02] can break the same variant with comparable data and time complexities but with a much lower memory complexity of 2^9. Consequently, the new technique did not improve the best previously known attack on 5 rounds.

In [BDK+18], authors greatly improved our attack, showing how to attack 5-round AES in data, memory and time complexities of less than $2^{22.5}$, which is about 500 times faster than any previous attack on the same variant. From practical verification, it turns out that the success rate of our full key recovery attack rose sharply from 0.24 to 1 as the amount of available data is increased from 2^{22} to 2^{23} in tiny increments of $2^{0.25}$.

By extending this technique to larger versions of AES, authors also obtained new attacks on AES-192 and AES-256 which have the best time complexity among all the attacks on 7-round AES which have practical data and memory complexities. In particular, by combining our attack with the *dissection technique* [DDKS12] and several other techniques, authors were able to beat this 18-year old record and to develop the best attacks on 7-round AES in this model. As a result, their attack on 7-round AES with 192-bit keys requires 2^{30} data, 2^{32} memory and 2^{153} time, which outperforms the Square attack in all three complexity measures simultaneously.

6.4. A new 5-round Secret-Key Distinguisher for AES

Using the 4-round "Mixture Differential" distinguisher based on Theorem 7 as starting point, we propose a way to extend it by 1 round at the end. As a result, we are able to set up a *new probabilistic 5-round secret-key distinguisher for AES which exploits a property which is independent of the secret key, of the details of the S-Box and of the MixColumns matrix* (expect for the branch number equal to 5). Even if such a distinguisher has higher complexity than the deterministic one presented in [GRR17], it can be used to set up a key-recovery attack on 6-round AES (better than a brute-force one) exploiting a distinguisher of the type [GRR17] - (initially) believed to be hard to exploit. It follows that this is *the first key-recovery attack for 6-round AES set up by a 5-round secret-key distinguisher for AES*. For completeness, since the 4-round distinguisher works also in the decryption direction, this new 5-round distinguisher - and the corresponding 6-round key-recovery attack - can also be set up in the reverse direction (i.e. using chosen ciphertexts instead of plaintexts).

6.4.1. Intersections of Subspaces and Useful Probabilities

Here we list some useful probabilities largely used in the following[11]. For our goal, we focus on the mixed space $\mathcal{M}$, but the same results can be easily generalized for the other subspaces $\mathcal{D}$, $\mathcal{C}$, $\mathcal{ID}$.

Let $I, J \subseteq \{0,1,2,3\}$. We recall that

$$\mathcal{M}_I \cap \mathcal{M}_J = \mathcal{M}_{I\cap J} \tag{6.1}$$

where $\mathcal{M}_I \cap \mathcal{M}_J = \{0\}$ if $I \cap J = \emptyset$.

For the follow-up, we also recall that given the events $A_1, \ldots, A_n$ in a probability space $(\Omega, \mathcal{F}, \mathbb{P})$

$$Prob\biggl(\bigcup_{i=1}^{n} A_i\biggr) = \sum_{k=1}^{n}\biggl((-1)^{k-1} \sum_{J\subset\{1,\ldots,n\},\,|J|=k} Prob\bigl(\bigcap_{j\in J} A_j\bigr)\biggr), \tag{6.2}$$

[11] We mention that the following probabilities are "sufficiently good" approximations for our scope, that is the errors of these approximations can be considered negligible for our scope. We clarify this claim in the following.

where the last sum runs over all possible sets J of cardinality k.

Proposition 9. *The probability $p_{|I|}$ that a random text x belongs to the subspace $\mathcal{M}_I$ for a certain $I \subseteq \{0,1,2,3\}$ with $|I| = l$ fixed is well approximated by*

$$p_{|I|} = Prob\big[\exists I \subseteq \{0,1,2,3\}\, |I| = l \;\; s.t. \;\; x \in \mathcal{M}_I\big] = (-1)^{|I|} \cdot \sum_{i=4-|I|}^{3} (-1)^i \cdot c_{|I|,i} \cdot \binom{4}{i} \cdot 2^{-32\cdot i} \tag{6.3}$$

where $c_{2,3} = 3$ and $c_{|I|,i} = 1$ for $\{|I|, i\} \neq \{2,3\}$.

Proof. Using the inclusive/exclusion principle (6.2) and due to (6.1), it follows that for $|I| = 1$

$$Prob\big[\exists I \subseteq \{0,1,2,3\}\, |I| = 1 \text{ s.t. } x \oplus y \in \mathcal{M}_I\big] = \sum_{I\subseteq\{0,1,2,3\},\, |I|=1} Prob(x \oplus y \in \mathcal{M}_I) = 4 \cdot 2^{-96}.$$

For $|I| = 3$ and using the law of total probability (4.11), the probability is given by:

$$\begin{aligned} &Prob\big[\exists I \subseteq \{0,1,2,3\}\, |I| = 3 \text{ s.t. } x \in \mathcal{M}_I\big] = \\ =&\sum_{j=1}^{3} \sum_{I\subseteq\{0,1,2,3\},\, |I|=j} (-1)^{j+1} \cdot Prob\big[x \in \mathcal{M}_I\big] = 4 \cdot 2^{-32} - 6 \cdot 2^{-64} + 4 \cdot 2^{-96}, \end{aligned}$$

since given 4 different sets $\mathcal{M}_I$ for $|I| = 3$ there are $\binom{4}{2} = 6$ possible intersections of 2 sets and $\binom{4}{3} = 4$ possible intersections of 3 sets (all intersections are not empty).

Finally, for $|I| = 2$ and using the law of total probability (4.11)

$$\begin{aligned} &Prob\big[\exists I \subseteq \{0,1,2,3\}\, |I| = 2 \text{ s.t. } x \in \mathcal{M}_I\big] = \\ =&\sum_{j=1}^{2} \sum_{I\subseteq\{0,1,2,3\},\, |I|=j} (-1)^{j} \cdot Prob\big[x \in \mathcal{M}_I\big] = 6 \cdot 2^{-64} - 12 \cdot 2^{-96}, \end{aligned}$$

since given 6 different sets $\mathcal{M}_I$ for $|I| = 2$ there are $\binom{6}{2} = 15$ possible intersections of 2 sets. However, note that only 12 of them are not empty (since $\mathcal{M}_{0,1} \cap \mathcal{M}_{2,3} = \mathcal{M}_{0,2} \cap \mathcal{M}_{1,3} = \mathcal{M}_{0,3} \cap \mathcal{M}_{1,2} = \emptyset$).

Since $\binom{6}{1} = \binom{4}{2} = 6$ and $\binom{6}{2} - 3 = \binom{4}{3} \cdot 3 = 12$, we obtain the desired result. □

Proposition 10. *Let x, y be two random elements. Assume that there exists $I \subseteq \{0,1,2,3\}$ such that $x \oplus y \in \mathcal{M}_I$ ($x \oplus y \notin \mathcal{M}_L$ for all $L \subseteq \{0,1,2,3\}$ with $|L| < |I|$). The probability that $\exists J \subseteq \{0,1,2,3\}$ with $|J| = l$ fixed such that $R(x) \oplus R(y) \in \mathcal{M}_J$ is well approximated by*

$$\begin{aligned} p_{|J|,|I|} \equiv\, & Prob\big[\exists J \subseteq \{0,1,2,3\}\, |J| = l \;\; s.t. \;\; R(x) \oplus R(y) \in \mathcal{M}_J \,\big|\, x \oplus y \in \mathcal{M}_I\big] = \\ &= (-1)^{|J|} \cdot \sum_{i=4-|J|}^{3} (-1)^i \cdot c_{|J|,i} \cdot \binom{4}{i} \cdot 2^{-8\cdot i\cdot |I|}. \end{aligned} \tag{6.4}$$

where $c_{2,3} = 3$ and $c_{|J|,i} = 1$ for $\{|J|, i\} \neq \{2,3\}$.

Proof. As before, for $|J| = 3$:

$$\begin{aligned} &Prob\big[\exists J \subseteq \{0,1,2,3\}\, |J| = 3 \text{ s.t. } R(x) \oplus R(y) \in \mathcal{M}_J \,|\, x \oplus y \in \mathcal{M}_I\big] = \\ =&\sum_{j=1}^{3} \sum_{J\subseteq\{0,1,2,3\},\, |J|=j} (-1)^{j+1} \cdot Prob\big[R(x) \oplus R(y) \in \mathcal{M}_J \,|\, x \oplus y \in \mathcal{M}_I\big] = \\ =&(-1)^3 \cdot \sum_{i=1}^{3} (-1)^i \cdot \binom{4}{i} \cdot 2^{-8\cdot i\cdot |I|} = 4 \cdot 2^{-8\cdot|I|} - 6 \cdot 2^{-16\cdot|I|} + 4 \cdot 2^{-24\cdot|I|}. \end{aligned}$$

By simple computation, it is possible to obtain similar results for $|J| = 2$ and $|J| = 1$, q.e.d. □

Proposition 11. *Let x, y be two random elements such that $x \oplus y \notin \mathcal{M}_I$ for each $I \subseteq \{0,1,2,3\}$. Then, the probability that $\exists J \subseteq \{0,1,2,3\}$ for $|J| = l$ fixed such that $R(x) \oplus R(y) \in \mathcal{M}_J$ is well approximated by*

$$\hat{p}_{|J|,3} \equiv Prob\big[\exists J \subseteq \{0,1,2,3\} \text{ s.t. } R(x) \oplus R(y) \in \mathcal{M}_J \,|\, x \oplus y \notin \mathcal{M}_I \, \forall I\big] = \frac{p_{|J|} - p_{|J|,3} \cdot p_3}{1 - p_3}. \quad (6.5)$$

Proof. Let A and B be two events, and let C be the event such that $A \cup C$ is equal to the sample space and such that $A \cap C = \emptyset$. Due to the law of total probability:

$$Prob(B) = Prob(B \,|\, A) \cdot Prob(A) + Prob(B \,|\, C) \cdot Prob(C).$$

Thus

$$\begin{aligned} p_{|J|} \equiv Prob\big[\exists J \subseteq \{0,1,2,3\} \text{ s.t. } R(x) \oplus R(y) \in \mathcal{M}_J\big] = \\ = Prob\big[\exists J \text{ s.t. } R(x) \oplus R(y) \in \mathcal{M}_J \,|\, x \oplus y \notin \mathcal{M}_I \, \forall I\big] \cdot Prob\big[x \oplus y \notin \mathcal{M}_I \, \forall I\big] + \\ + Prob\big[\exists J \text{ s.t. } R(x) \oplus R(y) \in \mathcal{M}_J \,|\, \exists I \text{ s.t. } x \oplus y \in \mathcal{M}_I\big] \cdot Prob\big[\exists I \text{ s.t. } x \oplus y \in \mathcal{M}_I\big]. \end{aligned}$$

Note that[12]

$$Prob\left[\exists I \text{ s.t. } x \oplus y \in \mathcal{M}_I\right] = Prob\left[x \oplus y \in \bigcup_{\forall I \subseteq \{0,1,2,3\}} \mathcal{M}_I\right] = Prob\left[x \oplus y \in \bigcup_{I \subseteq \{0,1,2,3\},\, |I|=3} \mathcal{M}_I\right] \equiv p_3.$$

It follows that

$$p_{|J|} = p_{|J|,3} \cdot p_3 + \hat{p}_{|J|,3} \cdot (1 - p_3),$$

q.e.d. □

Proposition 12. *Let x and y such that $x \oplus y \notin \mathcal{M}_I$ for each $I \subseteq \{0,1,2,3\}$. Then, the probability that $\exists J \subseteq \{0,1,2,3\}$ with $|J| = l$ fixed and $|I| + |J| \le 4$ such that $R^2(x) \oplus R^2(y) \in \mathcal{M}_J$ is well approximated by*

$$\tilde{p}_{|J|,3} \equiv Prob\big[\exists J \subseteq \{0,1,2,3\} \text{ s.t. } R^2(x) \oplus R^2(y) \in \mathcal{M}_J \,\big|\, x \oplus y \notin \mathcal{M}_I\big] = \frac{p_{|J|}}{1 - p_3}.$$

Proof. Remember that

$$Prob\big[\exists J \text{ s.t. } R^2(x) \oplus R^2(y) \in \mathcal{M}_J \,|\, \exists I \text{ s.t. } x \oplus y \notin \mathcal{M}_I\big] = 0.$$

Since

$$\begin{aligned} Prob\big[\exists J \subseteq \{0,1,2,3\} \text{ s.t. } R^2(x) \oplus R^2(y) \in \mathcal{M}_J\big] = \\ = Prob\big[\exists J \text{ s.t. } R^2(x) \oplus R^2(y) \in \mathcal{M}_J \,\big|\, x \oplus y \notin \mathcal{M}_I \, \forall I\big] \cdot Prob\big[x \oplus y \notin \mathcal{M}_I \, \forall I\big] + \\ + Prob\big[\exists J \text{ s.t. } R^2(x) \oplus R^2(y) \in \mathcal{M}_J \,\big|\, \exists I \text{ s.t. } x \oplus y \in \mathcal{M}_I\big] \cdot Prob\big[\exists I \text{ s.t. } x \oplus y \in \mathcal{M}_I\big] \end{aligned}$$

and using the same argumentation as before, it follows that

$$p_{|J|} = \tilde{p}_{|J|,3} \cdot (1 - p_3),$$

q.e.d. □

To provide a numerical example, if $|I| = |J| = 3$ the previous probabilities are well approximated by

$$p_3 = 2^{-30} - 3 \cdot 2^{-63} + 2^{-94}, \qquad p_{3,3} = 2^{-22} - 3 \cdot 2^{-47} + 2^{-70}$$
$$\hat{p}_{3,3} = 2^{-30} - 2\,043 \cdot 2^{-63} + 390\,661 \cdot 2^{-94} + \ldots$$

where p_3 and $\hat{p}_{3,3}$ are usually approximated by 2^{-30} and $p_{3,3}$ by 2^{-22}.

[12] If $x \oplus y \in \mathcal{M}_I$ for $|I| < 3$, then $\exists J$ with $|J| = 3$ and $I \subseteq J$ such that $x \oplus y \in \mathcal{M}_J$.

Remark. *All these probabilities are not the exact ones, but "good enough" approximations useful for our scope.* Here we give more details about this statement.

Firstly, consider the following concrete example. Consider the probability that a pair of texts t^1 and t^2 belong to the same coset of $\mathcal{M}_I$. This probability is usually approximated by $Prob(x \in \mathcal{M}_I) = 2^{-32\cdot(4-|I|)}$. On the other hand, in order to set up a (truncated) differential attack, one is interested to the case $t^1 \neq t^2$ (equivalently, $x \neq 0$). Thus, the "correct" probability should be

$$Prob(x \in \mathcal{M}_I \,|\, x \neq 0) = \frac{2^{32\cdot|I|} - 1}{2^{128} - 1} = 2^{-32\cdot(4-|I|)} - 2^{-128} + 2^{-128-32\cdot(4-|I|)} + ...$$

Secondly, we also remark that *the assumption behind the probabilities just given is that the elements x and y are uniform distributed, or (at least) very close to be uniform distributed*[13]*. In particular, we emphasize that this assumption is satisfied for all the events considered in the following to set up distinguishers and key-recovery attacks on 5- and 6-round AES.*

Number of Pairs

As last thing, we show that given texts in the same cosets of $\mathcal{C}_I$ (and similar for $\mathcal{M}_I$) for $I \subseteq \{0, 1, 2, 3\}$, the number of pairs of texts with n equal "generating variable(s) in $(\mathbb{F}_{2^8})^{|I|}$" for $0 \leq n \leq 3$ (as discussed at the end of Sect. 5.1.1) is given by

$$\binom{4}{n} \cdot 2^{32\cdot|I|-1} \cdot (2^{8\cdot|I|} - 1)^{4-n} \tag{6.6}$$

The proof of this formula for the case $|I| = 1$ is given in Sect. 5.2.1 – the formula for the other cases can be obtained in an analogous way.

6.4.2. 5-round *Probabilistic Mixture Differential* Secret-Key Distinguisher

Given n (plaintexts, ciphertexts), the idea is to divide them in sets such that particular relations hold among the variables that define the pairs of plaintexts that lie in the same set (similar to before). The distinguisher that we are going to present exploits the following property:

- consider *the number of sets for which two ciphertexts of at least one pair lie in the same subspace $\mathcal{M}_J$ for $|J| = 3$* (in other words, the number of sets for which two ciphertexts of at least one pair are equal in one anti-diagonal - if the final MixColumns operation is omitted). If the sets are properly defined, it is possible to prove that this number of sets *is a little lower for 5-round AES than for a random permutation, independently of the secret key.*

This property allows to set up a new distinguisher which is independent of the secret key, of the details of the S-Box and of the MixColumns matrix, and a new key-recovery attack on 6-round. In the following, we give all the details.

Details of the 5-round "Probabilistic Mixture Diff." Distinguisher

Consider 2^{32} chosen plaintexts with one active column (4 active bytes), e.g. a coset of $\mathcal{C}_0$, and the corresponding ciphertexts after 5-round. For each $(x_0, x_1), (y_0, y_1) \in \mathbb{F}_{2^8}^2$ such that $x_0 \neq y_0$ and $x_1 \neq y_1$, let $\mathcal{S}^{0,1}_{(x_0,x_1),(y_0,y_1)}$ be the set of pairs of plaintexts be defined as follows

$$\begin{aligned}\mathcal{S}^{0,1}_{(x_0,x_1),(y_0,y_1)} = \Bigg\{ (p, q) \in \mathbb{F}_{2^8}^{4\times4} \times \mathbb{F}_{2^8}^{4\times4} \,\Bigg|\, & p \equiv (x_0, x_1, A, B), q \equiv (y_0, y_1, A, B) \\ \text{or} \quad p \equiv & (x_0, y_1, A, B), q \equiv (y_0, x_1, A, B) \quad \text{for each } A, B \in \mathbb{F}_{2^8} \Bigg\}.\end{aligned}$$

[13] We refer to [Gra17b, App. A] for a concrete example of wrong probabilities when this assumption is *not* satisfied.

In other words, the pairs of plaintexts $p, q \in \mathcal{C}_0 \oplus a$ in $\mathcal{S}^{0,1}_{(x_0,x_1),(y_0,y_1)}$ are of the form

$$p \equiv a \oplus \begin{bmatrix} x_0 & 0 & 0 & 0 \\ x_1 & 0 & 0 & 0 \\ A & 0 & 0 & 0 \\ B & 0 & 0 & 0 \end{bmatrix} \qquad q \equiv a \oplus \begin{bmatrix} y_0 & 0 & 0 & 0 \\ y_1 & 0 & 0 & 0 \\ A & 0 & 0 & 0 \\ B & 0 & 0 & 0 \end{bmatrix},$$

or

$$p \equiv a \oplus \begin{bmatrix} x_0 & 0 & 0 & 0 \\ y_1 & 0 & 0 & 0 \\ A & 0 & 0 & 0 \\ B & 0 & 0 & 0 \end{bmatrix} \qquad q \equiv a \oplus \begin{bmatrix} y_0 & 0 & 0 & 0 \\ x_1 & 0 & 0 & 0 \\ A & 0 & 0 & 0 \\ B & 0 & 0 & 0 \end{bmatrix}.$$

Similar definitions can be given for the set $\mathcal{S}^{i,j}_{(x_0,x_1),(y_0,y_1)}$ for $i \neq j$, where the active bytes are in row i and j. Given 2^{32} plaintexts as before, it is possible to construct $\frac{1}{2^{17}} \cdot 6 \cdot 2^{31} \cdot (2^8-1)^2 \simeq 2^{32.574}$ different sets (the number of pairs of texts with 2 equal generating variables is given by formula (6.6)), where each set contains exactly 2^{17} different pairs of plaintexts (we emphasize that these pairs of plaintexts are not independent, in the sense that a particular relationship - among the generating variables - holds).

Consider $n \gg 1$ sets, and count the number of sets that contain *at least* one pair of plaintexts for which the corresponding ciphertexts (generated by 5-round AES or by a random permutation) belong to the same coset of a subspace $\mathcal{M}_J$ for $J \subseteq \{0,1,2,3\}$ and $|J| = 3$. As we are going to prove, this number is on average *smaller* for 5-round AES than for a random permutation, independently of the secret key, of the details of the S-Box and of the MixColumns matrix. In more details, the numbers of sets that satisfy the required property for 5-round AES - denoted by n_{AES} - and for a random permutation - denoted by n_{rand} - are well approximated by

$$n_{AES} \simeq n \cdot p_{AES} \qquad n_{rand} \simeq n \cdot p_{rand}$$

where

$$p_{AES} \simeq 2^{-13} - 524\,287 \cdot 2^{-46} \underbrace{-22\,370\,411\,853 \cdot 2^{-77}}_{\approx -\mathbf{2.604} \cdot \mathbf{2^{-44}}} + \ldots$$

$$p_{rand} \simeq 2^{-13} - 524\,287 \cdot 2^{-46} \underbrace{+45\,812\,722\,347 \cdot 2^{-77}}_{\approx +\mathbf{5.333} \cdot \mathbf{2^{-44}}} + \ldots$$

Even if the difference between the two probabilities is small, it is possible to distinguish the two cases with probability higher than 95% if the number n of sets $\mathcal{S}^{i,j}_{(x_0,x_1),(y_0,y_1)}$ - $\mathcal{S}$ for simplicity - satisfies $n \geq 2^{71.243}$.

In the following, we prove this result (which has been practically tested on a small scale AES) and we give all the details about the data and the computational costs.

Similarity with "classical" Truncated Differential Attack. Before going on, we emphasize the similarity with the 3-round truncated differential distinguisher [Knu94]. In that case, the idea is to count the number of pairs of texts that satisfy the truncated differential trail. In particular, given pairs of plaintexts in the same coset of a diagonal space $\mathcal{D}_i$, one counts the number of pairs for which the corresponding ciphertexts belong in the same coset of a mixed space $\mathcal{M}_J$ for $|J| = 3$. Since the probability of this event is higher for an AES permutation than for a random one[14], one can distinguish the two cases simply counting the number of pairs that satisfy the previous property. The idea of our disitinguisher is similar. However, instead of working on single pairs, one works with particular sets $\mathcal{S}$ of pairs and counts the number of sets for which at least one pair satisfies the (given) differential trail.

[14] Remember that this probability is approximately equal to 2^{-6} for the AES case and 2^{-30} for the random case (if J is not fixed – $|J| = 3$).

Proof - ***5-round AES***

As first thing, we prove the results just given, starting with the 5-round AES case.

Initial Considerations - 5-round AES. Our 5-round distinguisher is based on Theorem 7. Given plaintexts in the same coset of $\mathcal{C}_0$ and for a fixed $J \subseteq \{0, 1, 2, 3\}$, each set $\mathcal{S}^{i,j}_{(x_0,x_1),(y_0,y_1)}$ just defined has the following property after 4 rounds:

1. for each pair, the two texts after 4-round belong to the same coset of $\mathcal{M}_I$;

2. for each pair, the two texts after 4-round do not belong to the same coset of $\mathcal{M}_I$.

In other words, for a given set $\mathcal{S}_{(x_0,x_1),(y_0,y_1)}$, it is not possible that the two texts of some - not all - pairs belong to the same coset of $\mathcal{M}_J$ after 4-round and others not, while this can happen for a random permutation.

What is the probability of the two previous events for an AES permutation? Given a set $\mathcal{S}^{i,j}_{(x_0,x_1),(y_0,y_1)}$, the probability that the two texts of each pair belong to the same coset of $\mathcal{M}_J$ after 4-round is approximately 2^{-30}.

To prove this fact, let the event $\mathcal{E}^r_i$ be defined as following.

Definition 15. *Let $J \subseteq \{0, 1, 2, 3\}$ be fixed. Given a set $\mathcal{S}_{(x_0,x_1),(y_0,y_1)}$, we define $\mathcal{E}^r_i$ as the event that the i-th pair of $\mathcal{S}_{(x_0,x_1),(y_0,y_1)}$ for $i = 1, 2, ..., 2^{17}$ belong to the same coset of $\mathcal{M}_J$ after r rounds.*

In the following, let $\overline{\mathcal{E}^r_i}$ be the complementary event of $\mathcal{E}^r_i$. It follows that

$$\begin{aligned} Prob(\mathcal{E}^4_1 \wedge \mathcal{E}^4_2 \wedge ... \wedge \mathcal{E}^4_{2^{17}}) = Prob(\mathcal{E}^4_1) \cdot Prob(\mathcal{E}^4_2 \wedge ... \wedge \mathcal{E}^4_{2^{17}} \,|\, \mathcal{E}^4_1) = \\ = Prob(\mathcal{E}^4_1) \equiv p_3 = 2^{-30} - 3 \cdot 2^{-63} + 2^{-94}, \end{aligned}$$

where p_3 is defined as in (6.3). Indeed, note that $Prob(\mathcal{E}^4_i \,|\, \mathcal{E}^4_1) = 1$ for each $i = 2, ..., 2^{17}$ since if two texts of one pair belong (or not) to the same coset of $\mathcal{M}_J$ after 4 rounds, then the texts of all the other pairs have the same property. We remark again that this is due to the way in which the sets $\mathcal{S}$ are defined/constructed.

Using these initial considerations as starting point, we analyze in detail our proposed 5-round distinguisher.

1*st* Case. As we have just seen, two texts of all the pairs of each set belong to the same coset of a subspace $\mathcal{M}_I$ for $|I| = 3$ *after 4-round* with probability $p_3 \simeq 2^{-30}$. In other words, on average there are $2^{-30} \cdot n$ sets $\mathcal{S}$ such that the two texts of all the pairs belong to the same coset of a subspace $\mathcal{M}_J$ for $|J| = 3$ *after 4-round.*

Let $|J| = 3$. Since $Prob\big[R(x) \oplus R(y) \in \mathcal{M}_J \,|\, x \oplus y \in \mathcal{M}_I\big] = p_{3,3} \simeq 2^{-22}$ (see (6.4) for details) and since each set is composed of 2^{17} different pairs, the probability that the two ciphertexts of at least one pair of $\mathcal{S}$ belong to the same coset of $\mathcal{M}_J$ for $|J| = 3$ *after 5 rounds* is well approximated by

$$1 - \big(1 - \hat{p}_{3,3}\big)^{2^{17}} = 1 - \left(1 - \frac{p_3 \cdot (1 - p_{3,3})}{1 - p_3}\right)^{2^{17}} = 2^{-13} - 526\,327 \cdot 2^{-46} + ...$$

where $\hat{p}_{3,3}$ is defined in (6.5).

2*nd* Case. In the same way, the two texts of all the pairs of each set do not belong to the same coset of a subspace $\mathcal{M}_J$ for $|J| = 3$ *after 4-round* with probability $1 - p_3 \simeq 1 - 2^{-30}$. In other words, on average there are $(1 - 2^{-30}) \cdot n$ sets $\mathcal{S}$ such that the two ciphertexts of all the pairs of each set do not belong to the same coset of a subspace $\mathcal{M}_J$ for $|J| = 3$ *after 4-round.*

Let $|J| = 3$. Since $Prob(R(x) \oplus R(y) \in \mathcal{M}_J \,|\, x \oplus y \notin \mathcal{M}_I) = \hat{p}_{3,3} \simeq 2^{-30}$ (see (6.5) for details) and since each set is composed of 2^{17} different pairs, the probability that the two texts of at least one pair of $\mathcal{S}$ belong to the same coset of $\mathcal{M}_J$ for $|J| = 3$ *after 5 rounds* is well approximated by

$$1 - \big(1 - p_{3,3}\big)^{2^{17}} = 2^{-5} - 524\,287 \cdot 2^{-30} + 45\,812\,722\,347 \cdot 2^{-53} + ...$$

Final Result. The desired result is finally obtained using the *law* (or formula) *of total probability*

$$Prob(A) = \sum_i Prob(A \,|\, B_i) \cdot Prob(B_i)$$

which holds for each event A such that $\bigcup_i B_i$ is the *sample space*, i.e. the set of all the possible outcomes.

Given a set $\mathcal{S}$, the probability that two ciphertexts c^1 and c^2 of at least one pair satisfy the required property (i.e. $c^1 \oplus c^2 \in \mathcal{M}_J$ for $|J| = 3$) is given by

$$\begin{aligned} p_{AES} =& \big[1 - Prob(\overline{\mathcal{E}_1^5} \wedge \overline{\mathcal{E}_2^5} \wedge ... \wedge \overline{\mathcal{E}_{2^{17}}^5} \,|\, \mathcal{E}_i^4)\big] \cdot Prob(\mathcal{E}_i^4) + \big[1 - Prob(\overline{\mathcal{E}_1^5} \wedge \overline{\mathcal{E}_2^5} \wedge ... \wedge \overline{\mathcal{E}_{2^{17}}^5} \,|\, \overline{\mathcal{E}_i^4})\big] \cdot Prob(\overline{\mathcal{E}_i^4}) = \\ =& (1 - p_3) \cdot \left[1 - \left(1 - \frac{p_3 \cdot (1 - p_{3,3})}{1 - p_3}\right)^{2^{17}}\right] + p_3 \cdot \left[1 - \big(1 - p_{3,3}\big)^{2^{17}}\right] = \\ =& 2^{-13} - 524287 \cdot 2^{-46} - \underbrace{22\,370\,411\,853 \cdot 2^{-77}}_{\approx 2.604 \cdot 2^{-44}} + ... \end{aligned} \tag{6.7}$$

for a certain $i \in \{1, ..., 2^{17}\}$. Note that $Prob(\mathcal{E}_i^5 \wedge \mathcal{E}_j^5) = Prob(\mathcal{E}_i^5) \times Prob(\mathcal{E}_j^5)$ since the events $\mathcal{E}_i^5$ and $\mathcal{E}_j^5$ are independent for $i \neq j$.

Proof - *Random Permutation*

For a random permutation, given a set $\mathcal{S}$ defined as before, what is the probability that two ciphertexts - generated by a random permutation - of at least one pair satisfy the required property? By simple computation, such event occurs with (approximately) probability

$$\begin{aligned} p_{rand} =& 1 - \big(1 - p_3\big)^{2^{17}} = 1 - \big[1 - \big(2^{-30} - 3 \cdot 2^{-63} + 2^{-94}\big)\big]^{2^{17}} = \\ =& 2^{-13} - 524\,287 \cdot 2^{-46} + \underbrace{45\,812\,722\,347 \cdot 2^{-77}}_{\approx 5.333 \cdot 2^{-44}} + ... \end{aligned} \tag{6.8}$$

Remark. Before going on, we emphasize again that while a “classical” truncated differential distinguisher counts the number of pairs of texts that satisfy a particular differential trail, in our case we consider the number of sets of texts for which at least one pair satisfies a particular differential trail. This implies *a difference between the probabilities* that the previous event occurs for a random permutation - p_{rand} - and for 5-round AES - p_{AES}.

6.4.3. Data and Computational Complexity

Data Complexity

Since the difference between the two probabilities

$$\frac{|n_{AES} - n_{rand}|}{n_{AES}} \simeq \frac{|n_{AES} - n_{rand}|}{n_{rand}} \ll 1$$

(where the number of sets that satisfy the required property for the AES case and for the random case are denoted respectively by n_{AES} and n_{rand}) is very small, *what is the minimum number of sets*

$\mathcal{S}$ (or equivalently of cosets $\mathcal{C}_I$) to guarantee that the distinguisher works with high probability? Our goal here is to derive a good approximation for the number of initial cosets of $\mathcal{C}_I$ that is sufficient to appreciate this difference with probability *prob*.

To solve this problem, note that given n sets $\mathcal{S}$ of 2^{17} pairs defined as before, the distribution probability of our model is simply described by a *binomial distribution*, as discussed in Sect. 4.6.1. To derive concrete numbers for our distinguisher and based on De Moivre-Laplace theorem, we approximate the binomial distribution with a normal one. Moreover, we can simply consider the difference of the two distributions, which is again a normal distribution. That is, given $X \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $Y \sim \mathcal{N}(\mu_2, \sigma_2^2)$, then $X - Y \sim \mathcal{N}(\mu, \sigma^2) = N(\mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2)$. Indeed, in order to distinguish the two cases, note that it is sufficient to guarantee that the number of sets that satisfy the required property in the random case is higher than for the 5-round AES case. As a result, the mean μ and the variance σ^2 of the difference between the AES distribution and the random one are given by:

$$\mu = n \cdot |p_{rand} - p_{AES}| \qquad \sigma^2 = n \cdot \big[p_{rand} \cdot (1 - p_{rand}) + p_{AES} \cdot (1 - p_{AES})\big].$$

Since the probability density of the normal distribution is $f(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, it follows that

$$prob = \int_{-\infty}^{0} \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}} \mathrm{d}x = \int_{-\infty}^{-\mu/\sigma} \frac{1}{\sqrt{2\pi}}\, e^{-\frac{x^2}{2}} \mathrm{d}x = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{-\mu}{\sigma\sqrt{2}}\right)\right],$$

where $\operatorname{erf}(x)$ is the error function, defined as the probability of a random variable with normal distribution of mean 0 and variance $1/2$ falling in the range $[-x, x]$. We emphasize that the integral is computed in the range $(-\infty, 0]$ since we work in the case in which the number of sets with the required property for AES is lower than for the random case.

To have a probability of success higher than *prob*, the number of sets n has to satisfy:

$$n > \frac{2 \cdot [p_{rand} \cdot (1 - p_{rand}) + p_{AES} \cdot (1 - p_{AES})]}{(p_{rand} - p_{AES})^2} \cdot \Big[\operatorname{erfinv}\big(2 \cdot prob - 1\big)\Big]^2.$$

where $\operatorname{erfinv}(x)$ is the inverse error function. For the case $p_{rand}, p_{AES} \ll 1$, a good approximation of n is given by[15]

$$n > \frac{4 \cdot \max(p_{rand}, p_{AES})}{(p_{rand} - p_{AES})^2} \cdot \Big[\operatorname{erfinv}\big(2 \cdot prob - 1\big)\Big]^2. \tag{6.9}$$

Data Cost. First of all, given a single coset of a column space $\mathcal{C}_I$ for $|I| = 1$, the number of different pairs with two equal generating variables is given by $6 \cdot 2^{16} \cdot 2^{15} \cdot (2^8 - 1)^2 \simeq 2^{49.574}$ (see Eq. (6.6)), while the number of sets $\mathcal{S}$ that one can construct is well approximated by $2^{49.574}/2^{17} \simeq 2^{32.574}$.

For a probability of success of approximately 95% and since $|p_{AES} - p_{rand}| \simeq 2^{-41.01}$ and $p_{AES} \simeq p_{rand} \simeq 2^{-13}$, it follows that n must satisfy $n > 2^{71.243}$. Since a single coset of $\mathcal{C}_I$ for $|I| = 1$ contains approximately $2^{32.574}$ different sets $\mathcal{S}$, one needs approximately $2^{71.243} \cdot 2^{-32.574} \simeq 2^{38.669}$ different initial cosets of $\mathcal{C}_I$, that is approximately $2^{38.669} \cdot 2^{32} \simeq 2^{70.67}$ chosen plaintexts.

For completeness, we mention that it is possible to set up a modified version of this distinguisher that requires lower data (and computational) cost(s). In particular, in [Gra17b, App. D.2] we show that a similar distinguisher can be set up using only 2^{52} chosen plaintexts in the same initial coset of $\mathcal{C}_I$ with $|I| = 2$. Our choice to present a "less competitive" distinguisher is due to the fact that it will be the starting point for a key-recovery attack on 6-round, as shown in detail in the next section.

[15] Observe: $p_{rand} \cdot (1 - p_{rand}) + p_{AES} \cdot (1 - p_{AES}) < p_{rand} + p_{AES} < 2 \cdot \max(p_{rand}, p_{AES})$.

Computational Complexity

Here we discuss the computational cost for the case of cosets of $\mathcal{C}_I$ with $|I| = 1$. As for the 4-round distinguisher, a first possibility is to construct all the pairs, to divide them in sets $\mathcal{S}$ defined above, and to count the number of sets that satisfy the required property working on each set separately. Since just constructing all the pairs given $2^{38.67}$ cosets costs approximately of $2^{38.67} \cdot 2^{31} \cdot (2^{32} - 1) \simeq 2^{101.67}$ table look-ups, we present a more efficient way to implement the distinguisher. Before presenting the details, we highlight that the same analysis works also for modified version of the distinguisher – just presented – that work with plaintexts in the same initial coset of $\mathcal{C}_I$ with $|I| = 2$ (see [Gra17b, App. D.2] for more details). The computational cost of this modified version (that requires only 2^{52} chosen plaintexts) is well approximated by $2^{71.5}$ table look-ups or equivalently $2^{64.9}$ five-round encryptions.

Let $J \subseteq \{0, 1, 2, 3\}$ with $|J| = 3$. First of all, one has to re-order the ciphertexts with respect to a partial order $\preceq$ just defined. The cost of sorting a set of n texts w.r.t. a given partial order is $\mathcal{O}(n \cdot \log n)$ table look-ups.

For each coset of $\mathcal{C}_0$, *given ordered (plaintext, ciphertext) pairs and working only on consecutive ciphertexts*, the idea is to count the number of collisions for each set $\mathcal{S}^{i,j}_{(x_0,x_1),(y_0,y_1)}$. In more details, for each coset of $\mathcal{C}_0$ it is possible to construct $N = 3 \cdot 2^{15} \cdot (2^8 - 1)^2$ different sets $\mathcal{S}^{i,j}_{(x_0,x_1),(y_0,y_1)}$ for each $i, j \in \{0, 1, 2, 3\}$ with $i \neq j$ and for each $x_0 \neq y_0$ and $x_1 \neq y_1$. The idea is to consider a vector $A[0, ..., N - 1]$ such that the *i-th* component of such vector $A[i]$ contains the number of different pairs of one particular set $\mathcal{S}$ for which the two ciphertexts belong to the same coset of $\mathcal{M}_J$ for a certain J with $|J| = 3$. All the details are given in the following, while the pseudo-code is given in Algorithm 5.

To set up the distinguisher, it is sufficient to define a function φ that returns the index of a set $\mathcal{S}^{i,j}_{(x_0,x_1),(y_0,y_1)}$ (where $i < j$) in the vector $A[0, ..., N-1]$. First of all, assume that $x_0 < y_0$ and $x_1 < y_1$ (note that a set $\mathcal{S}$ contains all plaintexts generated by different combinations of these four variables, so this condition is always fulfilled). The function $\varphi(\cdot) : (\mathbb{F}_{2^8})^4 \times (\{0, 1, 2, 3\})^2 \to \mathbb{N}$ can be defined as[16]

$$\varphi(x_0, x_1, y_0, y_1, i, j) = 1\,065\,369\,600^{\phi(i,j)} \times \Phi(x_0, x_1, y_0, y_1) \tag{6.10}$$

where $1\,065\,369\,600 \equiv 2^{14} \cdot (2^8 - 1)^2$, where $\phi(0, 1) = 0$, $\phi(0, 2) = 1$, $\phi(0, 3) = 2$, $\phi(1, 2) = 3$, $\phi(1, 3) = 4$, $\phi(2, 3) = 5$ and

$$\Phi(x_0, x_1, y_0, y_1) = \left[(y_0 - x_0 - 1) + \frac{511 \cdot x_0 - x_0^2}{2}\right] + 32\,640 \cdot \left[(y_1 - x_1 - 1) + \frac{511 \cdot x_1 - x_1^2}{2}\right]$$

where each value of $\mathbb{F}_{2^8}$ is replaced by its corresponding number in $\{0, 1, ..., 255\}$.

As a result, using Algorithm 5 to implement the distinguisher, the computational cost is well approximated by

$$4 \cdot \Bigg[2^{32} \cdot \log(2^{32}) \text{ (re-ordering process) } + \left(2^{32} + 2^{31}\right) \text{ (access to } (p^i, c^i) \text{ and to } A[\cdot] \text{ -}$$
$$\text{- increment number of collisions)} \Bigg] + \frac{1}{2^{18}} \cdot 6 \cdot 2^{16} \cdot (2^8 - 1)^2 \text{ (final “for”)} \simeq 2^{39.07}$$

table look-ups for each initial coset, where $\binom{2^{32}}{2} \cdot 2^{-32} \simeq 2^{31}$ is the average number of pairs such that the two ciphertexts belong to the same coset of $\mathcal{M}_J$ for J fixed with $|J| = 3$. Since the

[16] Since $x_0 < y_0$ holds, note that x_0 can not be equal to 0xFF. The number of different pairs (x_0, y_0) that satisfy this condition is $\sum_{i=0}^{255} i = 32\,640$. Indeed, if $x_0 = $ 0x0 then y_0 can take 255 different values (all values expect 0), if $x_0 = $ 0x1 then y_0 can take 254 different values (all values expect 0x0, 0x1) and so. Moreover, for a given $(x, x + 1)$ where $x \neq$ 0x00, the number of different pairs $(\tilde{x}, \tilde{y})$ such that (1) $\tilde{x} < x$ and $\tilde{x} < \tilde{y}$ is equal to $\frac{511 \cdot x - x^2}{2}$. Indeed, there are x different possible values of $\tilde{x}$ and there are $256 - \tilde{x}$ different values of $\tilde{y}$ for each given $\tilde{x}$, for a total of $\sum_{i=256-x}^{255} i = \frac{511 \cdot x - x^2}{2}$.

```
Data: $2^{32}$ plaintexts in 1 coset of $\mathcal{C}_0$ (e.g. $\mathcal{C}_0 \oplus a$) and corresponding ciphertexts after 5
      rounds
Result: Number of sets $\mathcal{S}$ such that two ciphertexts of at least one pair of plaintexts belong
        to the same coset of $\mathcal{M}_J$ for a certain $J$ with $|J| = 3$
Let $A[0, ..., N-1]$ be an array initialized to zero, where $N = 3 \cdot 2^{15} \cdot (2^8 - 1)^2$    // $A[i]$
 refers to the $i$-th set $\mathcal{S}$
for each $j$ from 0 to 3 let $J = \{0,1,2,3\} \setminus j$ ($|J| = 3$) do
    let $(p^i, c^i)$ for $i = 0, ..., 2^{32} - 1$ be the (plaintexts, ciphertexts) in $\mathcal{C}_0 \oplus a$;
    re-order this set of elements w.r.t. the partial order $\preceq$ defined in Def. 14;// $\preceq$ depends
     on $J$
    $i \leftarrow 0$;
    while $i < 2^{32} - 1$ do
        $j \leftarrow i$;
        while $c^j \oplus c^{j+1} \in \mathcal{M}_J$ do
            $j \leftarrow j + 1$;
        end
        for each $k$ from $i$ to $j$ do
            for each $l$ from $k+1$ to $j$ do
                if $p^k \oplus p^l \in D_I$ for a certain $|I| = 2$ ($p^k$ and $p^l$ have two equal generating
                 variables)          // necessary condition s.t. $p^k \oplus p^l \in \mathcal{S}^{x,y}$ for
                 $x, y \in \{0,1,2,3\}$ with $x \neq y$ then
                    $A[\varphi(p^k, p^l)] \leftarrow A[\varphi(p^k, p^l)] + 1$;        // $\varphi(p^k, p^l)$ defined in (6.10)
                     returns the index of the set $\mathcal{S}^{x,y}$ s.t. $p^k \oplus p^l \in \mathcal{S}^{x,y}$ - this
                     step can be improved if one considers ordered plaintexts -
                     see [Gra17b, App. F] for details
                end
            end
        end
        $i \leftarrow j + 1$;
    end
end
$n \leftarrow 0$;
for each $i$ from 0 to $N - 1$ do
    if $A[i] \neq 0$ then
        $n \leftarrow n + 1$;
    end
end
return $n$.
```

Algorithm 5: Given (plaintexts, ciphertexts) pairs in the same coset of $\mathcal{C}_0$, *this algorithm counts the number of sets $\mathcal{S}$ for which two ciphertext of at least one pair belong in the same coset of $\mathcal{M}_J$ for $|J| = 3$.*

attacker must use $2^{38.66}$ different initial cosets to have a probability of success higher than 95%, the *total computational cost* is of $2^{39.07} \cdot 2^{38.66} = 2^{77.73}$ table look-ups, or equivalently $2^{71.1}$ five-round encryptions.

6.4.4. Practical Verification on small scale AES

In order to have a practical verification of the proposed distinguisher (and of the following key-recovery attack), we have practically verified the probabilities p_{AES} and p_{rand} given above. In

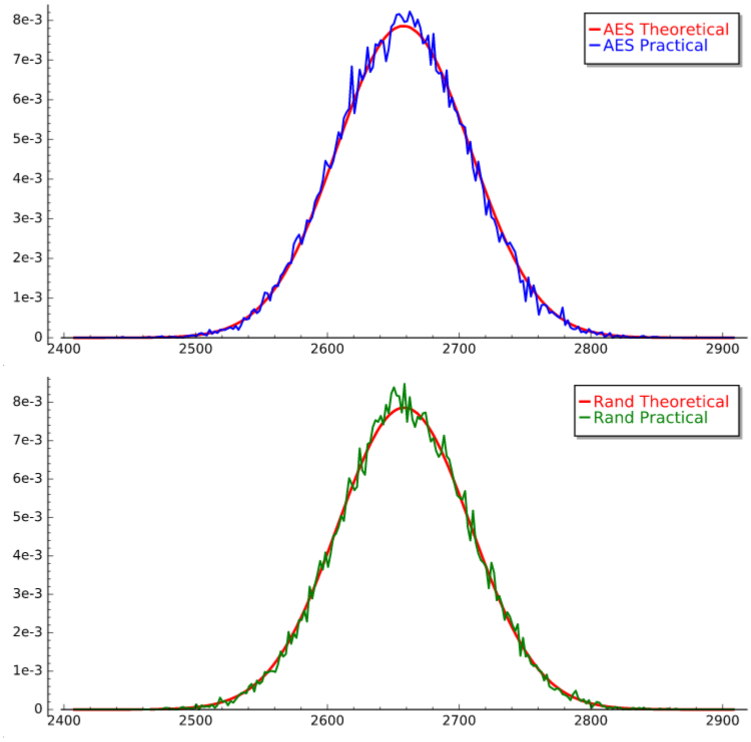


Figure 6.2.: *Probabilistic distributions of the number of collisions (i.e. number of pairs of ciphertexts in $\mathcal{S}$ that belong to the same coset of $\mathcal{M}_I$) for 5-round small scale AES and for a random permutation* - using 20 000 initial cosets.

particular, we verified them using a small scale AES, as proposed in [CMR05]. We emphasize that our verification on the small scale variant of AES is strong evidence for it to hold for the real AES, since the strategy used to theoretically compute such probabilities is independent of the fact that each word of AES is of 4 or 8 bits.

To compare the practical values with the theoretical ones, we list the theoretical probabilities p_{AES} and p_{rand} for the small scale case. First of all, for small scale AES the probabilities p_3 and $p_{3,3}$ are respectively equal to $p_3 = 2^{-14} - 3 \cdot 2^{-31} + 2^{-46}$ and $p_{3,3} = 2^{-10} - 3 \cdot 2^{-23} + 2^{-34}$.

W.l.o.g. we used cosets of $\mathcal{C}_0$ to practically test the two probabilities. Using the previous procedure and formula, the (approximately) probabilities that a set $\mathcal{S}$ satisfies the required property for 5-round small scale AES and for the random case are respectively

$$p_{AES} = 2^{-5} - 2\,047 \cdot 2^{-22} - \underbrace{221\,773 \cdot 2^{-37}}_{\approx 3.384 \cdot 2^{-21}} + \ldots$$

$$p_{rand} = 2^{-5} - 2\,047 \cdot 2^{-22} + \underbrace{698\,027 \cdot 2^{-37}}_{\approx 10.651 \cdot 2^{-21}} + \ldots$$

As a result, using formula (6.9) for $p_{rand} \simeq p_{AES} \simeq 2^{-5}$ and $|p_{rand} - p_{AES}| \simeq 2^{-17.19}$, it follows that $n \geq 2^{31.6}$ different sets $\mathcal{S}$ are sufficient to set up the distinguisher with probability higher than 95%.

Note that for small scale AES, a single coset of $\mathcal{C}_0$ contains 2^{16} (plaintexts, ciphertexts), or approximately $2^{15} \cdot (2^{16} - 1) \simeq 2^{31}$ different pairs. Since the number of pairs with two equal generating variables is given by $6 \cdot 2^8 \cdot 2^7 \cdot (2^4 - 1)^2 \simeq 2^{25.4}$ (also tested by computer test), it is possible to construct $3 \cdot 2^7 \cdot (2^4 - 1)^2 = 86400 \simeq 2^{16.4}$ sets $\mathcal{S}$ of 2^9 pairs. As a result, it follows that $2^{31.6} \cdot 2^{-16.4} = 2^{15.2}$ different initial cosets of $\mathcal{C}_0$ must be used, for a cost of $2^{47.2}$ chosen plaintexts.

For our tests, we used 2^{16} different initial cosets of $\mathcal{C}_0$ (keys used to encrypt the plaintexts in the AES case are randomly chosen and different for each coset - the key is not fixed). For each coset, we have used Algorithm 5 to count the number of sets $\mathcal{S}$ that satisfy the required property (i.e. the number of sets for which two ciphertexts of at least one pair are in the same coset of $\mathcal{M}_J$ for certain J with $|J| = 3$). As a result, for each initial coset $\mathcal{C}_0$ the (average) theoretical number of sets $\mathcal{S}$ that satisfy the required property for the random case - given by $n^T_{rand} = 86\,400 \cdot p_{rand}$ - and the (average) practical one found in our experiments - denoted by n^P_{rand} - are respectively:

$$n^T_{rand} \simeq 2\,658.27 \qquad\qquad n^P_{rand} \simeq 2\,658.23$$

Similarly, the (average) theoretical number of sets $\mathcal{S}$ that satisfy the required property for 5-round small scale AES - given by $n^T_{AES} = 86\,400 \cdot p_{AES}$ - and the (average) practical one found in our experiments - denoted by n^P_{AES} - are respectively:

$$n^T_{AES} \simeq 2\,657.69 \qquad\qquad n^P_{AES} \simeq 2\,657.65$$

In more details, the *total* numbers of sets $\mathcal{S}$ - for all the 2^{16} different initial cosets of $\mathcal{C}_0$ - that satisfy the required property for 5-round small scale AES and for a random permutation are given by

$$n^T_{rand} \simeq 174\,212\,383 \qquad\qquad n^T_{AES} \simeq 174\,174\,372$$
$$n^P_{rand} \simeq 174\,209\,761 \qquad\qquad n^P_{AES} \simeq 174\,171\,751$$

Note that the numbers of sets found in our experiments are close to the theoretical ones, and that the average number of sets for AES case is lower than for the random one, as predicted.

For completeness, the probabilistic distributions of the number of collisions for the AES and the random cases are given in Fig. 6.2. In both cases, the practical distribution is obtained using $20\,000 \equiv 2^{14.3}$ initial cosets. It is possible to observe that e.g. the theoretical variance matches the practical one in both cases.

6.5. Key-Recovery Attack on 6 rounds of AES-128

Using the previous distinguisher on 5-round AES (based on a property which is independent of the secret key) as starting point, we propose the first key-recovery attack on 6 rounds of AES that exploits a 5-round secret-key distinguisher. The strategy of the attack is similar to the one largely exploited by linear and differential cryptanalysis.

For the distinguisher just presented, the idea is to consider plaintexts in cosets of $\mathcal{C}_I$ for $I \subseteq \{0, 1, 2, 3\}$ with $|I| = 1$, construct all the possible pairs of two (plaintexts, ciphertexts) with two equal generating variables, divide them into sets $\mathcal{S}$ of 2^{17} pairs and count the number of sets for which two ciphertexts of at least one pair belong to the same coset of $\mathcal{M}_J$ for $|J| = 3$. To set up the key-recovery attack, the idea is simply to start with cosets of $\mathcal{D}_I$ for $I \in \{0, 1, 2, 3\}$, and to repeat the previous procedure for each guessed combination of the I-th diagonal of the secret key. *The crucial point is that the guessed 4-bytes of the key influence the way in which the pairs of texts are divided into the sets $\mathcal{S}$.* As a consequence, if the 4 guessed bytes are wrong (i.e. different from the right ones), the pairs are divided into sets $\mathcal{S}$ in a random way.

As we are going to prove, *for a wrongly guessed key the probability that a set $\mathcal{S}$ satisfies the required property* (that is, two ciphertexts of at least one pair belong to the same coset of $\mathcal{M}_J$) *is (approximately) equal to the probability of the random case p_{rand}, which is higher than the probability p_{AES} for the case of the right key.* As a result, the number of sets $\mathcal{S}$ for which two ciphertexts of at least one pair belong to the same coset of $\mathcal{M}_J$ for $|J| = 3$ is minimum for the right key. This allows to recover one diagonal of the secret key. In the following we present all the details.

Key-Recovery Attack - Details

Consider texts in a coset of $\mathcal{C}_I$ which is obtained by 1-round encryption of a coset of $\mathcal{D}_I$ with respect to a (partially) guessed key. Here we theoretically compute the probability that a set $\mathcal{S}$ satisfies the required property (that is, two ciphertexts of at least one pair belong to the same coset of $\mathcal{M}_J$) when the guessed key is not the right one. In other words, we are going to show that the behavior in the case of a wrongly guessed key (in the following denoted by "AES with a wrong key") is similar to the one of a random permutation.

Observe that the main difference between "AES with a wrong key" and a random permutation is given by the possibility in the first case to study the distribution of the pairs after each round - note that for a random permutation it is meaningless to consider the distribution of the texts after (e.g.) one round. In particular, a coset of a diagonal space $\mathcal{D}_I$ is always mapped into a coset of a column space $\mathcal{C}_I$ after one round independently of the key. On the other hand, we stress that *the way in which the pairs are distributed in the sets $\mathcal{S}$ depends on the guessed key.*

Consider a key-recovery attack on 6-round AES

$$\mathcal{D}_I \oplus a \xrightarrow[KeyGuess]{R(\cdot)} \underbrace{\bigcup_{(\mathbf{x},\mathbf{y})} \mathcal{S}^{i,j}_{\mathbf{x},\mathbf{y}} \subseteq \mathcal{C}_I \oplus b \xrightarrow[\text{prob. 1}]{R(\cdot)} \mathcal{M}_I \oplus c \xrightarrow{R(\cdot)} \mathcal{D}_J \oplus a' \xrightarrow[\text{prob. 1}]{R^2(\cdot)} \mathcal{M}_J \oplus c' \xrightarrow{R(\cdot)} \mathcal{M}_K \oplus c''}_{\text{5-round Secret-Key Distinguisher of Sect. 6.4}}$$

and focus on the middle round $\mathcal{M}_I \oplus c \xrightarrow{R(\cdot)} \mathcal{D}_J \oplus a'$ for $|I| = 1$ and $|J| = 3$. Assume the guessed key is wrong, and consider one set $\mathcal{S}^{i,j}_{(x_0,x_1),(y_0,y_1)}$. For this set, the number of pairs that belong to the same coset of $\mathcal{M}_J$ after four rounds can take any possible value between 0 and 2^{17} (that is, 0, 1, 2, ... or 2^{17}). Indeed, since the pairs are divided in sets $\mathcal{S}^{i,j}_{(x_0,x_1),(y_0,y_1)}$ in a random way, it is not possible to guarantee that the number of pairs that belong to the same coset of $\mathcal{M}_J$ after 4 rounds is only 0 or 2^{17} (as for "AES with the right key").

Using the same calculation as before and for a wrongly guessed key, given a set $\mathcal{S}^{i,j}_{(x_0,x_1),(y_0,y_1)}$, the probability $p^{WrongKey}_{AES}$ that two texts of at least one pair belong to the same coset of $\mathcal{M}_K$ for a certain $|K| = 3$ after 6 rounds is given by

$$p^{WrongKey}_{AES} = \sum_{n=0}^{2^{17}} \binom{2^{17}}{n} \cdot p_3^n \cdot (1-p_3)^{2^{17}-n} \cdot \left[1 - \left(1 - p_{3,3}\right)^n \cdot \left(1 - \frac{p_3 \cdot (1-p_{3,3})}{1-p_3}\right)^{2^{17}-n}\right],$$

which is well approximated by

$$p^{WrongKey}_{AES} = 2^{-13} - 524\,287 \cdot 2^{-46} + 45\,812\,722\,347 \cdot 2^{-77} + ...$$

Note that this probability is approximately equal to the one of the random case (see (6.8) for details), while we remember that the probability for "AES with the right key" is

$$p_{AES} = 2^{-13} - 524\,287 \cdot 2^{-46} - 22\,370\,411\,853 \cdot 2^{-77} + ...$$

where the difference between these two probabilities is approximately $|p^{WrongKey}_{AES} - p_{AES}| \simeq 2^{-41.011}$.

Data and Computational Costs

Data Cost. Assume the goal is to discover the I-th diagonal of the key with probability higher than 95%. Equivalently, the goal is to guarantee that the number of sets $\mathcal{S}^{i,j}_{(x_0,x_1),(y_0,y_1)}$ that satisfy the required property is the lowest one for the right key with probability higher than 95%.

To compute the data cost, the idea is to use the same analysis proposed for the 5-round distinguisher in Sect. 6.4.3. In particular, since there are 2^{32} candidates for each diagonal of the keys, one has to guarantee that the number of sets $\mathcal{S}^{i,j}_{(x_0,x_1),(y_0,y_1)}$ that satisfy the previous required property is the lowest one for the right key with probability higher than $(0.95)^{2^{-32}}$ (note that the 2^{32} tests - one for each candidate - are all independent).

```
Data: $2^{40.77}$ cosets of $\mathcal{D}_0$ (e.g. $\mathcal{D}_0 \oplus a_i$ for $a_i \in \mathcal{D}_0^{\perp}$) and corresponding ciphertexts after 6 rounds
Result: 4 bytes of the secret key - $(k_{0,0}, k_{1,1}, k_{2,2}, k_{3,3})$
Let $N[0, ..., 2^{32} - 1]$ be an array initialized to zero; // $N[k]$ denotes the number of sets $\mathcal{S}$ that satisfy the required property for the key $k$
/* 1st Step : for each guessed key, count the number of sets $\mathcal{S}$ with the required property */
for each $\hat{k}$ from $(0x00, 0x00, 0x00, 0x00)$ to $(0xff, 0xff, 0xff, 0xff)$ do
    for each coset $\mathcal{D}_0 \oplus a_i$ do
        (partially) encrypt the $2^{32}$ plaintexts w.r.t. the guessed key $\hat{k}$;
        use Algorithm 5 to count the number $n$ of sets $\mathcal{S}$ that satisfy the required property;
        $N[\psi(\hat{k})] \leftarrow N[\psi(\hat{k})] + n$;    // where $\psi(\hat{k} \equiv (k_0, k_1, k_2, k_3)) = \sum_{i=0}^{3} k_i \cdot 2^{8 \cdot i}$
    end
end
/* 2nd Step : look for the key for which number of sets $\mathcal{S}$ is minimum */
$min \leftarrow N[0]$;    // minimum number of sets
$\delta \leftarrow (0x00, 0x00, 0x00, 0x00)$;
for each $\hat{k}$ from $(0x00, 0x00, 0x00, 0x00)$ to $(0xff, 0xff, 0xff, 0xff)$ do
    if $N[\varphi(\hat{k})] < min$ then
        $min \leftarrow N[\varphi(\hat{k})]$;
        $\delta \leftarrow \hat{k} \equiv (k_{0,0}, k_{1,1}, k_{2,2}, k_{3,3})$;
    end
end
return $\delta$ - candidate of $(k_{0,0}, k_{1,1}, k_{2,2}, k_{3,3})$
```

Algorithm 6: *6-round key-recovery attack on AES exploiting a 5-round secret-key distinguisher.* The goal of the attack is to find 4 bytes of the secret key. The remaining bytes (the entire key) are found by brute force.

Using formula (6.9), one needs approximately $2^{73.343}$ different sets $\mathcal{S}^{i,j}_{(x_0,x_1),(y_0,y_1)}$ for each candidate of the i-th diagonal of the key. Since it is possible to construct approximately $3 \cdot 2^{15} \cdot (2^8 - 1)^2 \approx 2^{32.574}$ different sets for each initial coset of $\mathcal{D}_I$, one needs approximately $2^{73.343} \cdot 2^{-32.573} = 2^{40.77}$ different initial cosets of $\mathcal{D}_I$ to discover the I-th diagonal of the key with probability higher than 95%, for a total cost of $2^{40.77} \cdot 2^{32} = 2^{72.77}$ chosen plaintexts.

When one diagonal of the key is found[17], due to the computational cost of this step we propose to find the entire key (i.e. the other three diagonals) using a brute force attack.

Computational Cost. In order to implement the attack, the idea is to use Algorithm 5 for each possible guessed key in order to count the number of sets $\mathcal{S}$ that satisfy the required property (i.e. two ciphertexts of at least one pair belong to the same coset of $\mathcal{M}_J$ for a certain J with $|J| = 3$). Since this number of sets is higher for a wrongly guessed key than for the right one, it is possible to recover the right candidate of the key.

An implementation of the attack is described by the pseudo-code given in Algorithm 6. To compute the computational cost, it is sufficient to re-consider the cost of the 5-round distinguisher. Given a coset of $\mathcal{C}_0$, the cost to count the number of sets $\mathcal{S}$ with the required property is $2^{39.1}$ table look-ups. This step is repeated for each one of the 2^{32} (partially) guessed key and for each one of the $2^{40.77}$ initial cosets of $\mathcal{D}_0$, for a cost of $2^{39.05} \cdot 2^{40.77} \cdot 2^{32} = 2^{111.82}$ table look-ups. Moreover, one needs

[17] For completeness, we mention that it is possible to (slightly) reduce the data cost by relaxing the property that the number of sets $\mathcal{S}$ that satisfy the required property is the lowest one for the right key.

to partially compute 1-round encryption for each possible guessed key and for each initial coset, for a cost of $4 \cdot 2^{32} \cdot 2^{40.77} \cdot 2^{32} = 2^{106.77}$ S-Box look-ups. As a result, the total cost of finding one diagonal of the key is well approximated by $2^{111.82}$ table look-ups, or equivalently $2^{104.92}$ six-round encryptions (under the assumption 20 table/S-Box look-ups $\approx$ 1-round encryption). The total cost of finding the entire key (using brute force on the last three diagonal) is of $2^{104.92} + 2^{96} = 2^{104.93}$ six-round encryptions.

7 AES with a Single Secret S-Box

A key-recovery attack is any adversary's attempt to recover the cryptographic key of an encryption scheme. As stated by the Kerckhoffs' Principle, one common assumption is that the security of a cryptosystem must lie in the choice of its keys only: everything else (including the algorithm itself) should be considered public knowledge. *What happens if part of the crypto-system is instead kept secret?*

This problem has been first introduced by Biryukov and Shamir [BS01; BS10], where authors studied the security of AES-like ciphers which contain alternate (secret) layers of invertible S-Boxes and (secret) affine mappings. In particular, an attack was presented on five layers (SASAS, where S stands for substitution and A stands for affine mapping) of this construction which finds all secret components (up to an equivalence). Using the terminology of "rounds" as in the AES, this version consists of two and a half rounds.

A part from this work, several other results regarding cryptanalysis of ciphers with secret S-Boxes have been presented in the literature. To cite some examples, Gilbert and Chauvaud [GC94] presented a differential attack on the cipher Khufu (an unbalanced Feistel cipher), while Vaudenay provided cryptanalysis of reduced-round variants of Blowfish [Vau96]. In [BV05], Baignères and Vaudenay studied the security of AES*, a SPN identical to AES except that fixed S-Boxes are replaced by random and independent permutations, and proved that this construction resists linear and differential cryptanalysis with 4 inner rounds only (despite the huge cumulative effect of multipath characteristics that is induced by the symmetries of AES). Most recently, the lightweight cipher PRESENT was cryptanalyzed by Borghof *et al.* [BKLT11] also in the (extreme) case in which the S-Boxes are chosen uniformly at random for each round. Finally, in [BBK14], authors considered the ASASA scheme in order to design public key or white-box constructions using symmetric cipher components.

Attacks on AES with a Single Secret S-Box - State of the Art

The Advanced Encryption Standard (AES) is an iterated block cipher using 10, 12, or 14 rounds depending on the key size of 128, 192, or 256 bits. *Here we focus on the cipher that is derived from the AES by replacing the S-Box with a secret 8-bit S-Box while keeping everything else unchanged.* If the choice of S-Box is made uniformly at random from all 8-bit S-Boxes, the size of the secret information increases from 128 - 256 bits (the key size in the AES) to $128 + \log_2(2^8!) = 1812$ and $256 + \log_2(2^8!) = 1940$ bits respectively.

First of all, we recall that a randomly chosen S-Box is likely to have good properties against differential and linear cryptanalysis, as shown in [OCo93]. In particular, it has been shown there that, for mappings chosen uniformly at random from the set of all m-bit bijective mappings, the expected value of the highest probability of a (non-trivial) differential characteristic is at most $2m/2^m$. In our case where $m = 8$, this means that for a randomly chosen 8-bit S-Box the expected maximum probability of a differential characteristic is $16/2^8 = 2^{-4}$. Since the number of active S-Boxes for four rounds of the AES is at least 25, one has an upper bound of the probability for any 4-round differential characteristic of 2^{-100}, and thus an upper bound for any 8-round differential characteristic of 2^{-200}. This is sufficient to conclude that differential cryptanalysis will not pose a threat to variants of the AES where the S-Box is replaced by a randomly chosen 8-bit S-Box.

Table 7.1.: *Comparison of attacks on round-reduced AES with secret S-Box.* Data complexity is measured in number of required chosen plaintexts/ciphertexts (CP/CC). Time complexity is measured in round-reduced AES encryption equivalents (E), in memory accesses (M) or XOR operations (XOR). Memory complexity is measured in plaintexts (16 bytes). The case in which the final MixColumns operation is omitted is denoted by "r.5 rounds", that is r full rounds and the final round. New attacks are in bold. Strategy 1 (S1) denotes an attack that requires to find the details of the S-Box, while Strategy 2 (S2) denotes an attack that find directly the key.

Attack	Rounds	S1	S2	Data	Computation	Memory	Reference
TrD	**2.5 - 3**		✓	**$2^{13.6}$ CP**	**$2^{13.2}$ XOR**	**small**	**[GRR16]**
I	**2.5 - 3**		✓	**$2^{19.6}$ CP**	**$2^{19.6}$ XOR**	**small**	**[GRR16]**
I	3.5 - 4	✓		2^{16} CC	$2^{17.7}$ E	2^{16}	[TKKL15]
I	3.5 - 4	✓		2^{16} CP	$2^{28.7}$ E	2^{16}	[TKKL15, Sect. 3.5]
TrD	**3.5 - 4**		✓	**2^{30} CP**	**2^{36} M $\approx 2^{29.7}$ E**	**2^{30}**	**[GRR16]**
I	4.5 - 5	✓		2^{40} CC	$2^{38.7}$ E	2^{40}	[TKKL15]
I	4.5 - 5	✓		2^{40} CP	$2^{54.7}$ E	2^{40}	[TKKL15, Sect. 3.5]
Mult-of-n	**4.5 – 5**		✓	**$2^{53.25}$ CP**	**$2^{59.25}$ M $\approx 2^{52.6}$ E**	**2^{16}**	**[Gra18a]**
Mult-of-n	**4.5 – 5**		✓	**$2^{53.6}$ CP**	**$2^{55.6}$ M $\approx 2^{48.96}$ E**	**2^{40}**	**[Gra18a]**
ImD	**4.5 – 5**		✓	**$2^{76.37}$ CP**	**$2^{81.54}$ M $\approx 2^{74.9}$ E**	**2^{8}**	**[Gra18a]**
ImD	4.5 - 5		✓	$2^{76.5}$ CC	$2^{80.5}$ M $\approx 2^{73.86}$ E	2^{32}	[HCGW18]
I	5		✓	2^{96} CP	2^{96} M $\approx 2^{89.36}$ E	small	[HCGW18]
I	5		✓	2^{128} CC	$2^{129.6}$ XOR	small	[SLG+16]
I	5.5 - 6	✓		2^{64} CP/CC	2^{90} E	2^{69}	[TKKL15]

TrD: Truncated Differential, I: Integral, ImD: Impossible Differential.

A similar result can be proved for linear cryptanalysis using the bounds of linear characteristics from [OCo94].

State of the Art. The attacks on AES with a single secret S-Box in the literature – that is, [BS01; BS10] and [TKKL15] – exploit the following strategy:

1. first, the attacker determines the secret S-Box up to additive constants (that is, S-Box$(x \oplus a) \oplus b$ for unknown a and b);
2. then she uses this knowledge and applies key-recovery attacks present in the literature to derive the whitening key.

The property used for this strategy is usually the integral one, which is independent of the details of the secret S-Box. In particular, given a set Λ of 2^8 plaintexts $\{p_i\}$ with one active byte, it is well known that the corresponding texts after 4-round AES (assuming the last MixColumns operation is omitted) satisfy the following condition

$$\bigoplus_i \text{S-Box}^{-1}\big(R^4(c_i) \oplus k\big) = 0$$

where k is the final (secret) key. Since this equation is invariant under any affine transformation $A(\cdot)$ (i.e. $\bigoplus_i z_i = 0$ iff $\bigoplus_i A(z_i) = 0$), authors can find the S-Box up to additive constants, i.e. S-Box$(x \oplus a) \oplus b$ for unknown a and b. Exploiting this information and a classical integral attack on 4 rounds, they are then able to find the whitening key up to 256 variants, that is $(k_0, k_0 \oplus k_1, ..., k_0 \oplus k_{15})$

(where k_i is the i-th byte of the whitening key) for unknown k_0. Variant of this attack can be set up for up to 6 rounds of AES.

7.1. New Attacks on AES with a single Secret S-Box

In all the previous works, an attacker must work both on the secret S-Box and on the secret key, that is she has to first find information on the secret S-Box in order to discover the secret key. Thus, a natural question arises: *Is it also possible to directly find the secret key without exploiting/discovering any information about the secret S-Box?* In [SLG+16] and in our papers [GRR16; Gra18a], authors showed that this is possible *if* an assumption on the MixColumns matrix is guaranteed. Using the subspace-trail framework, in the following we present a *generic* technique to discover *directly* (i.e. without working on the S-Box) the secret key of AES up to 2^{32} variants, and we show how it can be combined with a truncated differential attack, an impossible differential attack, an integral attack and the multiple-of-n property.

7.1.1. Idea of the Attack

The main idea of our attack on AES with a secret S-Box is the following. As we have seen, a coset of $\mathcal{D}_i$ is mapped into a coset of $\mathcal{C}_i$ after one round. Using some particular (but very common) properties of the MixColumns matrix, it is possible to choose a subset of a coset of $\mathcal{D}_i$ which depends on the secret key, such that it is mapped after one round into a subset of a coset of $\mathcal{D}_J \cap \mathcal{C}_i \subseteq \mathcal{D}_J$ with probability 1. That is, consider a subset of a coset of $\mathcal{D}_i$ which depends on the guessed values of some bytes of the secret key. If these guessed values are wrong, then after one round this subset of $\mathcal{D}_i$ is mapped into a subset of a coset of $\mathcal{C}_i$. Instead, if these guessed values are correct, then after one round this subset of $\mathcal{D}_i$ is mapped into a subset of a coset of $\mathcal{D}_J$ with probability 1. Note that also when the guessed values are wrong it is possible that the initial subset is mapped into a subset of a coset of $\mathcal{D}_J$ after one round, but this happens with probability strictly less than 1. Using this property together with other considerations, the attacker can identify the right key.

In more details, referring to Figure 7.1, consider a subset of a coset of $\mathcal{D}_i$ related to the guess secret key as plaintexts.

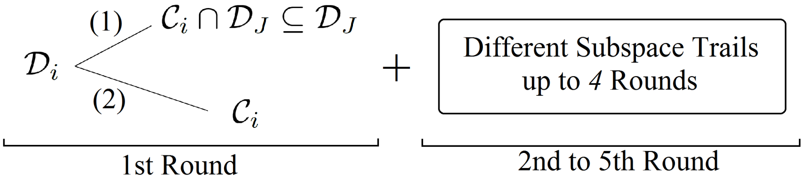

$\mathcal{D}_i$
(1)
$\mathcal{C}_i \cap \mathcal{D}_J \subseteq \mathcal{D}_J$
(2)
$\mathcal{C}_i$
+
Different Subspace Trails up to *4* Rounds
1st Round
2nd to 5th Round

Figure 7.1.: *Strategy of the attacks on AES with a secret S-Box.* Starting with a subset of a coset of $\mathcal{D}_i$ which depends on the guessed values of the secret key, it is mapped after one round into a subset of a coset of $\mathcal{D}_J$ if the guessed values is correct - case (1), or into a subset of a coset of $\mathcal{C}_i$ if the guessed values is wrong - case (2). As a consequence, the subspace trails up to the 5-*th* round are different for the two cases, and this allows to set up various key-recovery attacks.

If the guessed value is correct - case (1) of Fig. 7.1 (that is, if the difference of two consecutive-diagonal bytes of the plaintexts is equal to the difference of the same bytes of the secret key), then this set is mapped into a subset of a coset of $\mathcal{C}_i \cap \mathcal{D}_J \subseteq \mathcal{D}_J$ for a certain J with $|J| = 3$. If the guessed value is wrong - case (2) of Fig. 7.1, then this set is mapped into a subset of a coset of $\mathcal{C}_i$.

This attack exploits the following particular property of the MixColumns $n \times n$ matrix M:

given a row of M, there exists a subset of coefficients whose xor-sum is equal to zero, that is

$$\exists j \in \{0, 1, ..., n-1\} : \qquad \bigoplus_{i \in I} M_{j,i} = 0$$

where $I \subseteq \{0, 1, ..., n-1\}$.

Obviously, if two coefficients are equal (e.g. $M_{j,l} = M_{j,k}$ for $k \neq l$), then the previous property is always satisfied (e.g. for $I = \{l, k\}$).

7.1.2. Equal coefficients in MixColumns Matrix

As first case, we consider the case of two equal coefficients. Assume that a matrix M has two equal coefficients in the same row, e.g. $M_{i,j} = M_{i,k}$ for $j \neq k$.

As example, the AES MixColumns matrix satisfies this property, since e.g. $M_{0,2} = M_{0,3}$. Note that if a row of a circulant matrix satisfies this property, then all other rows satisfies it. We recall that a *cyclic/circulant matrix* is a matrix where each row vector is rotated one element to the right relative to the preceding row vector, that is:

$$circ(c_0, c_1, ..., c_{n-1}) = \begin{bmatrix} c_0 & c_1 & \cdots & c_{n-1} \\ c_{n-1} & c_0 & \cdots & c_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & \cdots & c_0 \end{bmatrix}.$$

Using this properties of M, our attack is based on the following lemma.

Lemma 7 ([GRR16]). *Let p^1 and p^2 two texts such that $p^1_{i,j} = p^2_{i,j}$ for each $(i, j) \neq \{(0, 0), (1, 1)\}$ and $p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1}$. If $p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1} = k_{0,0} \oplus k_{1,1}$ (where k is the secret key of the first round), then after one round they belong to the same coset of $\mathcal{C}_0 \cap \mathcal{D}_{0,1,3} \subseteq \mathcal{D}_{0,1,3}$, that is $R(p^1) \oplus R(p^2) \in \mathcal{C}_0 \cap \mathcal{D}_{0,1,3} \subseteq \mathcal{D}_{0,1,3}$.*

Proof. First of all, note that these two texts p^1 and p^2 belong in the same coset of $\mathcal{D}_0 \cap \mathcal{C}_{0,1} \subseteq \mathcal{D}_0$ (by definition of $\mathcal{D}_0$). As we have already seen, if two elements belong to the same coset of $\mathcal{D}_0$, then after one round they belong to the same coset of $\mathcal{C}_0$. Thus, it is sufficient to prove that $R(p^1) \oplus R(p^2) \in \mathcal{D}_{0,1,3}$.

Since $R(p^1) \oplus R(p^2) \in \mathcal{C}_0$, in order to prove that $R(p^1) \oplus R(p^2) \in \mathcal{D}_{0,1,3}$ it is sufficient to prove that $R(p^1)_{2,0} \oplus R(p^2)_{2,0} = 0$. By simple computation:

$$\begin{aligned} R(p^1)_{2,0} = {} & \text{S-Box}(p^1_{0,0} \oplus k^0_{0,0}) \oplus \text{ S-Box}(p^1_{1,1} \oplus k^0_{1,1}) \oplus \\ & \oplus \text{0x02} \cdot \text{ S-Box}(p^1_{2,2} \oplus k_{2,2}) \oplus \text{0x03} \cdot \text{ S-Box}(p^1_{3,3} \oplus k_{3,3}). \end{aligned}$$

First of all observe that S-Box$(p^1_{0,0} \oplus k^0_{0,0}) \oplus$ S-Box$(p^1_{1,1} \oplus k^0_{1,1}) = 0$. Indeed, since $p^1_{0,0} \oplus p^1_{1,1} = k_{0,0} \oplus k_{1,1}$ by definition, then $p^1_{0,0} \oplus k^0_{0,0} = p^1_{1,1} \oplus k^0_{1,1}$, that is S-Box$(p^1_{0,0} \oplus k^0_{0,0}) =$ S-Box$(p^1_{1,1} \oplus k^0_{1,1})$, or equivalently S-Box$(p^1_{0,0} \oplus k^0_{0,0}) \oplus$ S-Box$(p^1_{1,1} \oplus k^0_{1,1}) = 0$. Thus:

$$R(p^1)_{2,0} = \text{0x02} \cdot \text{ S-Box}(p^1_{2,2} \oplus k_{2,2}) \oplus \text{0x03} \cdot \text{ S-Box}(p^1_{3,3} \oplus k_{3,3})$$

and in a similar way:

$$R(p^2)_{2,0} = \text{0x02} \cdot \text{ S-Box}(p^2_{2,2} \oplus k_{2,2}) \oplus \text{0x03} \cdot \text{ S-Box}(p^2_{3,3} \oplus k_{3,3}).$$

Since $p^1_{2,2} = p^2_{2,2}$ and $p^1_{3,3} = p^2_{3,3}$ by definition, it follows that $R(p^1)_{2,0} = R(p^2)_{2,0}$, and so the thesis. □

Note that no information of the S-Box is used. As shown in the following, this property can be used to discover - directly - the secret key.

Previous Lemma can be easily generalized for each possible combination of consecutive-diagonal bytes.

Proposition 13 ([GRR16]). *Let p^1 and p^2 two texts such that*

$$p^1_{i,j} = p^2_{i,j} \qquad \forall (i,j) \neq \{(n,m),(k,l)\}$$

and

$$p^1_{k,l} \oplus p^1_{n,m} = p^2_{k,l} \oplus p^2_{n,m},$$

where $l - k \equiv_4 m - n$. If $p^1_{k,l} \oplus p^1_{n,m} = p^2_{k,l} \oplus p^2_{n,m} = k_{k,l} \oplus k_{n,m}$ (where k is the secret key of the first round), then after one round they belong to the same coset of $\mathcal{C}_{l-k} \cap \mathcal{D}_{\{0,1,2,3\}\setminus r} \subseteq \mathcal{D}_{\{0,1,2,3\}\setminus r}$ (the indexes are taken modulo 4), where r is defined as the row of the MixColumn matrix M such that $M_{r,n} = M_{r,k}$. Equivalently, $R(p^1) \oplus R(p^2) \in \mathcal{C}_{k-l} \cap \mathcal{D}_{\{0,1,2,3\}\setminus r}$.

Using the subspace trails of Sect. 4.3, this implies for example that:

- after 3 rounds, two plaintexts in the set V_δ are mapped into a subset of a coset of $\mathcal{M}_J$ with probability 1 in case (1), while this happens only with probability 2^{-8} - i.e. strictly less than 1 - in case (2);
- after 4 rounds, the probability that two texts in the previous set V_δ are mapped into the same coset of $\mathcal{M}_J$ is higher in case (1) - approximately 2^{-22} - than in case (2) - approximately 2^{-30};
- after 5 rounds, the probability that two texts in the previous set V_δ are mapped into the same coset of $\mathcal{M}_j$ is equal to zero in case (1), while is strictly different from zero in case (2) - approximately 2^{-94}.

"Weak" Secret-Key Distinguisher. Such a strategy has been introduced in [SLG+16] at Crypto 2016, in order to set up the first *"weak" secret-key distinguisher* on 5-round AES. A "weak" secret-key distinguisher can be seen as something in the middle between a key-recovery attack[1] and a secret-key distinguisher (which is independent of the secret key). The goal is to distinguish a random permutation from a block cipher, where *(1st)* it is sufficient to find only part of the key to achieve this goal and *(2nd)* it does not exploit any detail of the S-Box.

In order to construct the secret key distinguisher presented in [SLG+16], authors simply consider all the input-output space, and divide it in the 2^8 subsets $\tilde{V}_\Delta$ defined as $\tilde{V}_\Delta = \{(p,c) \,|\, c_{0,0} \oplus c_{1,3} = \Delta\}$ for each possible $\Delta \in \mathbb{F}_{2^8}$, and without any other assumptions on the other bytes. Note that $|\tilde{V}_\Delta| = 2^{120}$. Then, using the link between zero-correlation linear hulls and the integral/balance property, they are able to prove that for an AES permutation and for $\Delta = k_{0,0} \oplus k_{1,3}$ the sum of the plaintexts of the corresponding set $\tilde{V}_\Delta$ is equal to zero, that is the balance property holds[2]. Instead, for a random permutation, the probability that there exists one Δ with the previous property is only 2^{-120}. This distinguisher works only in the decryption direction (i.e. using chosen ciphertexts) and only if the final MixColumns operation is not omitted. Moreover, there is no evidence that this distinguisher can work with less than the entire input-output space. We refer to [SLG+16] for more details. To summarize, this distinguisher requires the full codebook (i.e. 2^{128} texts), and the verification cost is well approximated by 2^{128} XOR operations.

[1] We recall that any key-recovery attack is also a secret-key distinguisher, in the sense that it can be used to distinguish between a cipher and a random permutation. Obviously, a key-recovery attack used for this goal is not independent of the secret key, since the property used to distinguish the two cases is the existence of the key.

[2] In [SLG+16], authors presented also a similar distinguisher always based on balance property. In this case, the idea is to divide the entire input-output space in 2^{32} subsets $\tilde{W}_\Delta$ defined as $\tilde{W}_\Delta = \{(p,c) \,|\, c_{0,0} \oplus c_{1,3} = \delta_0, c_{0,1} \oplus c_{3,2} = \delta_1, c_{1,2} \oplus c_{2,1} = \delta_2, c_{2,0} \oplus c_{3,3} = \delta_3\}$, where $\Delta = (\delta_0, \ldots, \delta_3)$. Also in this case, for an AES permutation there exists one Δ for which the balance property holds among the plaintexts, while for a random permutation this happens only with probability 2^{-96}

7.1.3. A More Generic Strategy

Instead of exploiting the fact that two elements of each row of the MixColumns matrix M are equal, a more general property can be used to mount similar attacks, that is the fact that the XOR-sum of 2 or more elements of each row of M is equal to zero. That is, it is possible to set up an attack also in the case in which for each row r (or for some of them) of M there exists a set $J_r \subseteq \{0, 1, 2, 3\}$ such that

$$\bigoplus_{j \in J_r} M_{r,j} = 0 \tag{7.1}$$

As an example, each row of the AES MixColumns matrix M satisfies this condition, e.g. for the first row

$$M_{0,0} \oplus M_{0,1} \oplus M_{0,2} = \text{0x02} \oplus \text{0x03} \oplus \text{0x01} = 0, \quad M_{0,i} \neq M_{0,j} \ \forall i, j \in \{0, 1, 2\}.$$

As a special case, if two elements $M_{r,j}$ and $M_{r,k}$ of a row r are equal (that is $M_{r,j} = M_{r,k}$ for $j \neq k$), then the previous condition is obviously satisfied (vice-versa does not hold).

To explain how to exploit property (7.1), we show how to adapt the strategy just described to this case. As we have already said, the idea is to choose a set of plaintexts $\mathcal{A}_\delta$ which depends on a guessed key δ. When δ assumes the "right" value (which depends on the secret key), then the set $\mathcal{A}_\delta$ is mapped after one round into a coset of $\mathcal{D}_I$ for some I (where $|I| \leq 3$) with probability 1, while for other values of δ this happens only with probability strictly less than 1. Since the idea is to exploit the same strategy, we limit ourselves here to define the set $\mathcal{A}_\delta$ in the case in which a sum of elements of each row of M is equal to zero.

Proposition 14 ([Gra18a]). *Let M be the AES MixColumns matrix such that*

$$M_{i,0} \oplus M_{i,1} \oplus M_{i,2} = 0 \qquad i = \{0, 1\}.$$

Let p^1 and p^2 be two texts, s.t. $p^1_{i,j} = p^2_{i,j}$ for all $(i, j) \neq \{(0, 0), (1, 1), (2, 2)\}$ and

$$p^1_{i,j} \oplus p^1_{k,l} = p^2_{i,j} \oplus p^2_{k,l} \qquad \forall (i, j), (k, l) \in \{(0, 0), (1, 1), (2, 2)\} \text{ and } (i, j) \neq (k, l).$$

If $p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1} = k_{0,0} \oplus k_{1,1}$ and $p^1_{0,0} \oplus p^1_{2,2} = p^2_{0,0} \oplus p^2_{2,2} = k_{0,0} \oplus k_{2,2}$, then $R(p^1) \oplus R(p^2) \in \mathcal{C}_0 \cap \mathcal{D}_{2,3}$ with probability 1 (i.e. after one round, p^1 and p^2 belong to the same coset of $\mathcal{C}_0 \cap \mathcal{D}_{2,3}$). This happens with probability 2^{-16} in the other cases.

Proof. Note that the two plaintexts p^1 and p^2 belong to the same coset of $\mathcal{D}_0$. Since a coset of diagonal space $\mathcal{D}_I$ is always mapped after one round into a coset of a column space $\mathcal{C}_I$, after one round they belong to the same coset of $\mathcal{C}_0$ with probability 1. To prove the statement, it is sufficient to prove that $[R(p^1) \oplus R(p^2)]_{0,0} = [R(p^1) \oplus R(p^2)]_{1,0} = 0$.

By simple calculation

$$\begin{aligned} R(p^1)_{0,0} = &\text{0x02} \cdot \text{S-Box}(p^1_{0,0} \oplus k_{0,0}) \oplus \text{0x03} \cdot \text{S-Box}(p^1_{1,1} \oplus k_{1,1}) \oplus \\ &\oplus \text{S-Box}(p^1_{2,2} \oplus k_{2,2}) \oplus \text{S-Box}(p^1_{3,3} \oplus k_{3,3}). \end{aligned}$$

Since $p^1_{0,0} \oplus p^1_{1,1} = k_{0,0} \oplus k_{1,1}$, it follows that $\text{S-Box}(p^1_{0,0} \oplus k_{0,0}) = \text{S-Box}(p^1_{1,1} \oplus k_{1,1})$ and in a similar way $\text{S-Box}(p^1_{0,0} \oplus k_{0,0}) = \text{S-Box}(p^1_{2,2} \oplus k_{2,2})$. Since the sum of the first three elements is equal to zero, then $R(p^1)_{0,0} = \text{S-Box}(p^1_{3,3} \oplus k_{3,3})$, and similarly $R(p^2)_{0,0} = \text{S-Box}(p^2_{3,3} \oplus k_{3,3})$. Since $p^1_{3,3} = p^2_{3,3}$, it follows that $R(p^1)_{0,0} = R(p^2)_{0,0}$. The same argumentation holds also for $R(p^1)_{1,0} = R(p^2)_{1,0}$. □

This proposition can be easily generalized for a more generic MixColumns matrix M for which the sum of three or four coefficients are equal to zero. Moreover, given J fixed, if the sum $\bigoplus_{j \in J} M_{r,j}$ is equal to zero for more than a single row r, the following Lemma follows immediately.

Table 7.2.: *Practical Numbers for the case of Circulant Invertible Matrices.* The second column gives the number of invertible matrices MC for which MC or MC^{-1} has two equal coefficients in each row, while the third one gives the number of invertible matrices for which the sum of ≥ 2 the same row of MC or MC^{-1} is equal to zero.

$\mathbb{F}_{2^m}^{4\times 4}$	**Number Invertible Matrices**	**Two Equal Coeff.**	**Zero-Sum of ≥ 2 Coeff.**
$m = 4$	61 440	32 640 (53.125%)	45 600 (74.22%)
$m = 8$	4 278 190 080	165 550 080 (3.87%)	293 556 000 (6.87%)

Table 7.3.: *Practical Numbers for the case of Circulant MDS Matrices.* The second column gives the number of MDS matrices MC for which MC or MC^{-1} has two equal coefficients in each row, while the third one gives the number of MDS matrices for which the sum of ≥ 2 elements in the same row of MC or MC^{-1} is equal to zero.

$\mathbb{F}_{2^m}^{4\times 4}$	**Number MDS Matrices**	**Two Equal Coeff.**	**Zero-Sum of ≥ 2 Coeff.**
$m = 4$	16 560	10 080 (60.87%)	12 480 (75.36%)
$m = 8$	4 015 735 920	126 977 760 (3.16%)	249 418 560 (6.21%)

Lemma 8 ([Gra18a]). *Assume there exist $J \subseteq \{0, 1, 2, 3\}$ and $r, w \in \{0, 1, 2, 3\}$ with $r \neq w$ such that*

$$\bigoplus_{j\in J} M_{r,j} = \bigoplus_{j\in J} M_{w,j} = 0.$$

Let p^1 and p^2 defined as before. It follows that if $p^1_{j,j} \oplus p^1_{l,l} = p^2_{j,j} \oplus p^2_{l,l} = k_{j,j} \oplus k_{l,l}$ for each $j, l \in J$, then $p^1 \oplus p^2 \in \mathcal{C}_k \cap \mathcal{D}_{\{0,1,2,3\}\setminus\{r,w\}}$ with probability 1, otherwise this happens in general with prob. 2^{-16}.

To prove this lemma, it is sufficient to exploit the previous proposition and to observe that if two plaintexts belong to the same coset of $\mathcal{C}_k \cap \mathcal{D}_{\{0,1,2,3\}\setminus\{r\}}$ and of $\mathcal{C}_k \cap \mathcal{D}_{\{0,1,2,3\}\setminus\{w\}}$, then they belong to their intersections $\mathcal{C}_k \cap \mathcal{D}_{\{0,1,2,3\}\setminus\{r,w\}}$.

What is the number of matrices that satisfy condition (7.1) with respect to the number of matrices with two equal coefficients in each row? Since we consider AES-like ciphers, we limit ourselves to practical count[3] both these numbers for the cases of *circulant* matrices in $\mathbb{F}_{2^m}^{4\times 4}$ for $m = 4, 8$. We remember that the strategy just proposed works in the encryption direction if the MixColumns matrix satisfies one of the two previous properties and/or in the decryption direction if the inverse MixColumns matrix satisfies them. For this reason, we compute the number of MixColumns matrices for which one of the two previous properties is satisfied in the encryption direction (i.e. by MC) *or* in the decryption direction (i.e. by MC^{-1}).

In Table 7.2 we list our results limiting to consider invertible matrices, while in Table 7.3 we list our results limiting to consider MDS (Maximal Distance Separable) matrices. Observing the numbers in the tables, both for these two cases and both for $m = 4$ and $m = 8$, the number of matrices that satisfy condition (7.1) is (largely) higher than the number of matrices with two equal coefficients in each row. E.g. for the case $m = 8$, this number increases of 77.32% (e.g. $2^{27.3}$ vs $2^{28.13}$) for the invertible matrices case, and of 96.42% (e.g. $2^{26.92}$ vs $2^{27.89}$) for the MDS matrices case (that is, the number has doubled).

[3] The source codes are available at `https://github.com/Krypto-iaik/Attacks_AES_SecretSBox2`

7.2. Truncated Diff. Attacks up to 4-round AES with a Single Secret S-Box

7.2.1. Truncated Differential Attack on 3 rounds of AES with Secret S-Box

Here, we present an attack on 3 rounds of AES with a secret S-Box. The attack - illustrated in Fig. 7.2 - works as follows.

Consider a pair of plaintexts p^1 and p^2 with the following conditions:

$$\forall(i,j) \neq \{(0,0),(1,1)\} : p^1_{i,j} = p^2_{i,j} \quad \text{and} \quad p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1}.$$

As we have seen, if $p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1} = k_{0,0} \oplus k_{1,1}$, then p^1 and p^2 belong to the same coset of $\mathcal{D}_{0,1,3}$ after one round with probability 1. Consequently, after three rounds they belong to the same coset of $\mathcal{M}_{0,1,3}$ with probability 1 (or of $\mathcal{ID}_{0,1,3}$ if the final MixColumns is omitted), since a coset of $\mathcal{D}_{0,1,3}$ is mapped into a coset of $\mathcal{M}_{0,1,3}$ with probability 1.

Instead, if $p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1} \neq k_{0,0} \oplus k_{1,1}$, then p^1 and p^2 belong to the same coset of $\mathcal{D}_{0,1,3}$ after one round only with probability 2^{-8} (that is, only if $R(p^1)_{2,0} \oplus R(p^2)_{2,0} = 0$). Thus, after three rounds they belong to the same coset of $\mathcal{M}_{0,1,3}$ only with probability[4] 2^{-8}. Our attack exploits the fact that these probabilities are different in order to find $k_{0,0} \oplus k_{1,1}$.

The idea is to consider n different pairs of plaintexts (with one plaintext in common) for each possible value of δ, that is $n \cdot 2^8$ pairs of plaintexts p^1 and p^2 such that $p^1_{i,j} = p^2_{i,j}$ for each $(i,j) \neq \{(0,0),(1,1)\}$ and $p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1} = \delta$. Given a δ, the attacker checks if the corresponding n pairs of ciphertexts belong or not to the same coset of $\mathcal{M}_{0,1,3}$. If not, then the key is wrong due to previous considerations.

Data and Computational Costs. What is the probability that all the false key candidates are discarded (i.e. they do not pass the test) using n pairs for each δ? This probability is given by $1-(1-2^{-8})^{2^8 \cdot n} \simeq 1 - e^{-n}$. If $n = 3$ (that is, 4 chosen plaintexts - one plaintext is in common), then this probability is higher than 95%. Thus, in order to find 1 byte of the key, $4 \cdot 2^8 = 2^{10}$ chosen plaintexts. The cost of the attack can be approximated to $3 \cdot 2^8 = 2^{9.6}$ XOR operations (for 4 chosen plaintexts, the attacker computes only 3 XOR operations, since she considers only 3 different pairs).

In order to find the secret key, for each of the four diagonals, the attacker has to repeat the same attack for three consecutive-diagonal bytes differences of the same diagonal, as for example $k_{0,0} \oplus k_{1,1}$, $k_{1,1} \oplus k_{2,2}$ and $k_{2,2} \oplus k_{3,3}$ for the first diagonal (note that the difference $k_{0,0} \oplus k_{3,3}$ and all the other differences of these four bytes of the first diagonal are given by the sum of the previous ones). As result, the attacker is able to find the whitening key up to $(2^8)^4 = 2^{32}$ variants, if she does not use any information about the secret S-Box. Thus, the total cost of the attack is $12 \cdot 2^{10} = 2^{13.6}$ chosen plaintexts and $12 \cdot 2^{9.6} = 2^{13.2}$ XOR operations.

Without discovering any information about the secret S-Box, the attacker is able to find the secret key up to 2^{32} variants.

Practical Verification. The attack just presented has been practically verified[5]: here we report the practical results. Suppose that an attacker is looking for a byte difference, e.g. $\delta \equiv k_{0,0} \oplus k_{1,1}$ (similarly for the other cases). As we have seen, in order to have a probability of success higher than

[4] Given two random texts x and y in they same coset of $\mathcal{D}_0$, they belong to the same coset of $\mathcal{M}_{0,1,3}$ with probability 2^{-24} - see Theorem 5. However, in this case we are not considering random texts in $\mathcal{D}_0$, but two texts that belong after one round to the same coset of $\mathcal{D}_{0,1,3}$ with probability 2^{-8}. Since two texts belong to the same coset of $\mathcal{M}_{0,1,3}$ if and only if they belong to the same coset of $\mathcal{D}_{0,1,3}$ two rounds before, we obtain that the probability for the studied case is 2^{-8} and not 2^{-24}.

[5] The source codes of the attacks on AES with a secret S-Box in this section are available at `https://github.com/Krypto-iaik/Attacks_AES_SecretSBox`

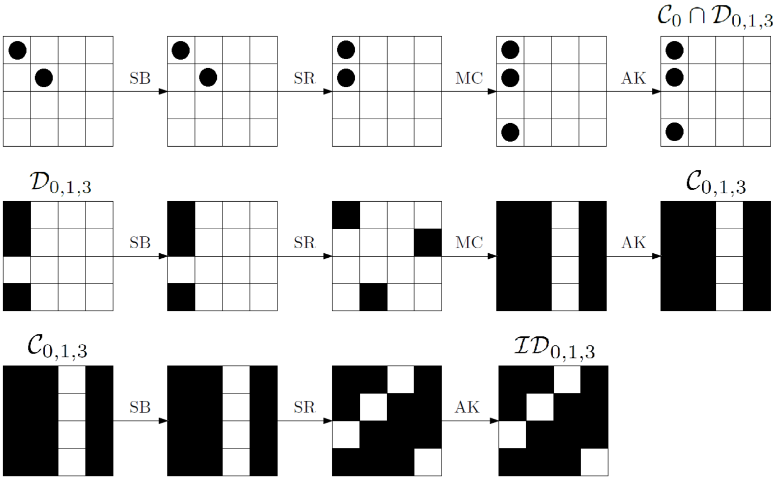


Figure 7.2.: *3-round Truncated Differential Attack on AES with secret S-Box.* The choice of the plaintexts (i.e. $p_{0,0} \oplus p_{1,1} = k_{0,0} \oplus k_{1,1}$) guarantees that after one round there are only three bytes with non-zero difference instead of four, that is the plaintexts belong to the same coset of $\mathcal{C}_0 \cap \mathcal{D}_{0,1,3}$. White box denotes denotes a byte with a zero-difference, while a black box denotes a byte with non-zero difference.

95%, she has to use 3 pairs of texts (with one text in common), and the computational cost can be approximated by $3 \cdot 2^8 = 768$ XOR operations (in the worse case). However, consider the case of a wrong guessed value of δ. In this case, when the attacker finds the first pair of ciphertexts that does not belong to the same coset of $\mathcal{M}_0$, she can immediately deduce that the guessed value δ is certainly wrong, without considering the other remaining pairs. For this reason, we can expect that the practical computational cost is lower than the theoretical one (which is computed analyzing the worse case). In effect, our practical results show that the *average* computational cost of the attacker is of 261 XOR operations, that is 1/3 of the theoretical one.

7.2.2. Integral Attack on 3 Rounds of AES with Secret S-Box

A similar technique works for the case of the square attack. As we have just seen, given two plaintexts p^1 and p^2 such that $p^1_{i,j} = p^2_{i,j}$ for each $(i, j) \neq \{(0, 0), (1, 1)\}$ and $p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1} = \delta$, then they belong to the same coset of $\mathcal{C}_0 \cap \mathcal{M}_{0,1,3}$ after one round if $\delta = k_{0,0} \oplus k_{1,1}$.

The idea of the attack is the following. Consider the set V_δ defined as in (7.2):

$$V_\delta = \{(p^i, c^i) \text{ for } i = 0, ..., 2^8-1 \,|\, \forall i : p^i_{0,0} \oplus p^i_{1,1} = \delta \quad \text{and} \quad \forall (k, l) \neq \{(0, 0), (1, 1)\},\ i \neq j : p^i_{k,l} = p^j_{k,l}\},$$

If $\delta = k_{0,0} \oplus k_{1,1}$, one round encryption of V_δ corresponds to

$$\begin{bmatrix} x & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & x \oplus \delta & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} \rightarrow \begin{bmatrix} y \oplus \tilde{c}_{0,0} & \tilde{c}_{0,1} & \tilde{c}_{0,2} & \tilde{c}_{0,3} \\ \text{0x03} \cdot y \oplus \tilde{c}_{1,0} & \tilde{c}_{1,1} & \tilde{c}_{1,2} & \tilde{c}_{1,3} \\ \tilde{c}_{2,0} & \tilde{c}_{2,1} & \tilde{c}_{2,2} & \tilde{c}_{2,3} \\ \text{0x02} \cdot y \oplus \tilde{c}_{3,0} & \tilde{c}_{3,1} & \tilde{c}_{3,2} & \tilde{c}_{3,3} \end{bmatrix}$$

for each $x \in \mathbb{F}_{2^8}$, where $y = \text{S-Box}(x \oplus k_{0,0})$. That is, if $\delta = k_{0,0} \oplus k_{1,1}$, then the set V_δ is mapped into $\mathcal{D}_{0,1,3} \cap \mathcal{C}_0 \cap \mathcal{M}_3$, which implies that the bytes in positions $(0, 0), (1, 0)$ and $(3, 0)$ can take each possible values in $\mathbb{F}_{2^8}$. Instead, if $\delta \neq k_{0,0} \oplus k_{1,1}$, no claims can be made about the bytes of the first column (the others are obviously constant). Equivalently, these two cases correspond to:

$$\begin{bmatrix} A & C & C & C \\ A & C & C & C \\ C & C & C & C \\ A & C & C & C \end{bmatrix}, \qquad \begin{bmatrix} ? & C & C & C \\ ? & C & C & C \\ ? & C & C & C \\ ? & C & C & C \end{bmatrix},$$

respectively for $\delta = k_{0,0} \oplus k_{1,1}$ and $\delta \neq k_{0,0} \oplus k_{1,1}$, where A, B, C and ? denote active/balance/constant/unknown byte (see Sect. 3.3.1 for details). Since

$$\begin{bmatrix} A & C & C & C \\ A & C & C & C \\ C & C & C & C \\ A & C & C & C \end{bmatrix} \rightarrow \begin{bmatrix} B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \end{bmatrix}$$

after 2 rounds, it follows that the 3-round encryption of V_δ has the balance property if $\delta = k_{0,0} \oplus k_{1,1}$. Instead, if $\delta \neq k_{0,0} \oplus k_{1,1}$, the probability that V_δ satisfies the balance property after 3-round if $\delta \neq k_{0,0} \oplus k_{1,1}$ is $(2^{-8})^{-16} = 2^{-128}$, since it is not in general possible to guarantee any property of the one round of encryption of V_δ.

Thus, the idea is to consider 2^8 different sets V_δ, one for each possible values of δ, and to check if the balance property on the ciphertexts is satisfied or not. If the balance property is not satisfied, then the value δ as candidate for $k_{0,0} \oplus k_{1,1}$ is certainly wrong. What is the probability that all the false candidates do not satisfy this test? By simply computation is $(1 - 2^{-128})^{2^8 - 1} \simeq 1 - 2^{-120}$. As a result, in order to find one byte of the secret key, the data complexity is $2^8 \cdot 2^8 = 2^{16}$, while the computational complexity can be approximated to 2^{16} XOR operations.

As for the previous attack, the idea is to repeat the attack for three different consecutive-diagonal bytes, and for all the four diagonals. In this way, the attacker is able to find 2^{32} variants of the whitening key without working on the secret S-Box. The total data complexity for the attack can be approximated to $12 \cdot 2^{16} = 2^{19.6}$ chosen plaintexts, and a cost of $2^{19.6}$ XOR operations.

Practical Verification. As last thing, we report that the practical computational cost of this attack is approximately the same of the theoretical one, and that the given data complexity allows to have an high probability of success, as indicated above.

7.2.3. Truncated Differential Attack on 4-round AES with a single Secret S-Box

As for the case of 3-round of AES, we present an attack on 4-round of AES with secret S-Box which exploits the truncated differential attack and the subspace trail. The previous truncated differential attack on 3-round of AES with secret S-Box exploits a subspace trail with probability 1. In this section, we present a truncated differential to attack 4-round of AES with secret S-Box - illustrated in Fig. 7.3 - that exploits the subspace trail described in Sect. 4.3.2, which has probability strictly less than 1 (but greater than 0).

The idea of the attack is to exploit the fact that two elements that belong to the same coset of $\mathcal{D}_I$ belong to the same coset of $\mathcal{M}_J$ after three rounds with probability higher than two random elements. To set up the attack, we exploit this fact together with the possibility to map a subset of a coset of $\mathcal{D}_i$ (which depends on the secret key) into a subset of a coset of $\mathcal{D}_I$ after one round.

Consider two plaintexts p^1 and p^2 such that $p^1_{i,j} = p^2_{i,j}$ for each $(i,j) \neq \{(0,0),(1,1)\}$ and $p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1} = \delta$. As we have seen, if $\delta = k_{0,0} \oplus k_{1,1}$, then the two plaintexts belong to the same coset of $\mathcal{D}_{0,1,3}$ after one round. First of all, the previous choice of plaintexts can be generalized.

Proposition 15 ([GRR16]). *Let p^1 and p^2 be two texts such that*

$$p^1_{i,j} = p^2_{i,j} \qquad \forall (i,j) \neq \{(0,0),(0,3),(1,1),(1,2),(2,0),(2,3),(3,1),(3,2)\}$$

and

$$p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1} = \delta_0, \qquad p^1_{1,2} \oplus p^1_{2,3} = p^2_{1,2} \oplus p^2_{2,3} = \delta_1,$$
$$p^1_{0,3} \oplus p^1_{3,2} = p^2_{0,3} \oplus p^2_{3,2} = \delta_2, \qquad p^1_{2,0} \oplus p^1_{3,1} = p^2_{2,0} \oplus p^2_{3,1} = \delta_3.$$

Then, if $\delta_0 = k_{0,0} \oplus k_{1,1}$, $\delta_1 = k_{1,2} \oplus k_{2,3}$, $\delta_2 = k_{0,3} \oplus k_{3,2}$ and $\delta_3 = k_{2,0} \oplus k_{3,1}$ (where k is the secret key of the first round), then after one round they belong to the same coset of $\mathcal{C}_0 \cap \mathcal{D}_{0,1,3} \subseteq \mathcal{D}_{0,1,3}$, that is $R(p^1) \oplus R(p^2) \in \mathcal{C}_0 \cap \mathcal{D}_{0,1,3}$.

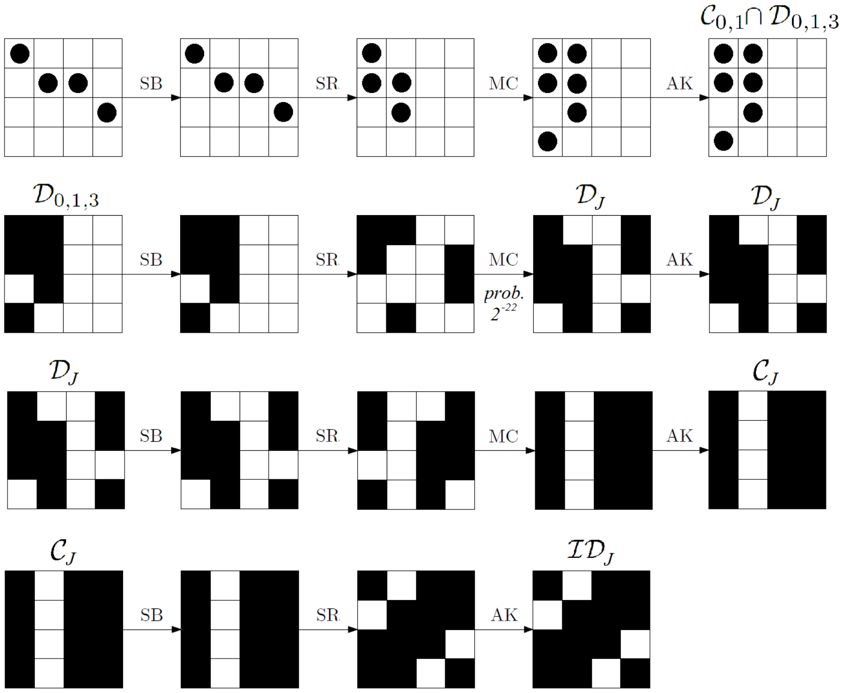


Figure 7.3.: *4-round Truncated Differential Attack on AES with secret S-Box.* The choice of the plaintexts (i.e. $p_{0,0} \oplus p_{1,1} = k_{0,0} \oplus k_{1,1}$ and $p_{1,2} \oplus p_{2,3} = k_{1,2} \oplus k_{2,3}$) guarantees that after one round they belong to the same coset of $\mathcal{C}_{0,1} \cap \mathcal{D}_{0,1,3}$. As a consequence, after three rounds they belong to the same coset of $\mathcal{C}_J$ for a certain $J \subseteq \{0, 1, 2, 3\}$ with $|J| = 3$ with probability 2^{-22} instead of 2^{-30}. White box denotes denotes a byte with a zero-difference, while a black box denotes a byte with non-zero difference.

The proof is similar to that of Lemma 13.

In the following, we consider pairs of plaintexts such that $p^1_{i,j} = p^2_{i,j}$ for each $(i, j) \neq \{(0, 0), (1, 1), (1, 2), (2, 3)\}$ and $p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1} = \delta_0$, $p^1_{1,2} \oplus p^1_{2,3} = p^2_{1,2} \oplus p^2_{2,3} = \delta_1$. If $\delta_0 = k_{0,0} \oplus k_{1,1}$ and $\delta_1 = k_{1,2} \oplus k_{2,3}$, these two plaintexts belong to the same coset of $\mathcal{D}_{0,1,3}$ with probability 1, that is they belong to the same coset of $\mathcal{M}_{0,1,3}$ after three rounds with probability 1. Using Prop. 5, this means that if $\delta_0 = k_{0,0} \oplus k_{1,1}$ and $\delta_1 = k_{1,2} \oplus k_{2,3}$, the two plaintexts belong to the same coset of $\mathcal{C}_J$ with probability $(2^8)^{-12+4\cdot|J|}$ after three rounds. That is, if $|J| = 3$, $\delta_0 = k_{0,0} \oplus k_{1,1}$ and $\delta_1 = k_{1,2} \oplus k_{2,3}$, then after three rounds these two plaintexts belong to the same coset of $\mathcal{C}_J$ with probability 2^{-24} for a fixed J with $|J| = 3$, or with probability $4 \cdot 2^{-24} = 2^{-22}$ for a free J with $|J| = 3$. Since a coset of $\mathcal{C}_J$ is mapped into a coset of $\mathcal{M}_J$ with probability 1, we can conclude that if $\delta_0 = k_{0,0} \oplus k_{1,1}$ and $\delta_1 = k_{1,2} \oplus k_{2,3}$, the two plaintexts belong to the same coset of $\mathcal{M}_J$ for a certain J with $|J| = 3$ with probability 2^{-22}.

Consider now the case $\delta_0 \neq k_{0,0} \oplus k_{1,1}$ or/and $\delta_1 \neq k_{1,2} \oplus k_{2,3}$. In this case, we can consider the corresponding ciphertexts as randomly distributed, that is they belong to the same coset of $\mathcal{M}_J$ for a certain J with $|J| = 3$ after four rounds only with probability $4 \cdot 2^{-32} = 2^{-30}$.

Remark. Before to go on, we explain why it is not possible to work on a single δ (as for the previous attack on 3 rounds of Sect. 7.2.1). Suppose to consider two plaintexts p^1 and p^2 such that $p^1_{i,j} = p^2_{i,j}$ for each $(i, j) \neq \{(0, 0), (1, 1)\}$ and $p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1} = \delta$. By definition, these two plaintexts belong to the same coset of $\mathcal{D}_0 \cap \mathcal{C}_{0,1} \subseteq \mathcal{D}_0$, independently by δ. Thus, as shown in Sect. 4.3.3, the probability that they belong to the same coset of $\mathcal{M}_J$ for $|J| = 3$ is zero - see probability (4.6) - independently of δ.

Data and Computational Cost

Data Complexity. The idea of the attack is to exploit these different probabilities in order to recover the key. In particular, consider $n \leq 2^{16}$ plaintexts defined as before for each possible values of δ_0 and δ_1. If $\delta_0 = k_{0,0} \oplus k_{1,1}$ and $\delta_1 = k_{1,2} \oplus k_{2,3}$, then we expect approximately $n \cdot (n-1) \cdot 2^{-23}$ different pairs of ciphertexts that belong to the same coset of $\mathcal{M}_J$ for a certain J with $|J| = 3$ (i.e. collisions), while $n \cdot (n-1) \cdot 2^{-31}$ collisions if $\delta_0 \neq k_{0,0} \oplus k_{1,1}$ or/and $\delta_1 \neq k_{1,2} \oplus k_{2,3}$. For example, if $n = 2^{16}$, we expect on average $2^8 = 256$ collisions for the first case and 2 in the other one. By our experiments, we check that $n = 2^{13}$ is (largely) sufficient to find the right value of $k_{0,0} \oplus k_{1,1}$ and $k_{1,2} \oplus k_{2,3}$.

Assume that $k_{0,0} \oplus k_{1,1}$ and $k_{1,2} \oplus k_{2,3}$ have been found. The idea is to proceed in the same way to find the 2^{32} variants of the whitening secret key. To improve the total cost of the attack, suppose that for each diagonal the attacker one difference of two consecutive-diagonal bytes is known. A good strategy is to use it to find the others. As an example, a good strategy could be to use the knowledge of $\delta_0 = k_{0,0} \oplus k_{1,1}$ and to look for $\delta_2 = k_{0,3} \oplus k_{3,2}$ (similar for δ_0 and $\delta_3 = k_{2,0} \oplus k_{3,1}$), instead to work on δ_2 and δ_3. Note that both the method allows to find the secret key, but in the first case the attacker does $2 \cdot 2^8 = 2^9$ tests, while in the second one 2^{16}. Thus, the data complexity cost can be approximated by $8 \cdot 2^8 \cdot 2^8 + 2 \cdot (2^8)^2 \cdot 2^{13} = 2^{30}$ chosen plaintexts.

Computational Complexity. First of all, observe that the first step of the attack (i.e. to find δ_0 and δ_1) is the most expensive one. Thus, the total cost can be approximated by this step (which is repeated two times - on the first and on the second diagonals, and then on the third and on the fourth diagonal). Moreover, note that the attacker must construct all the possible pairs of ciphertexts and check if they belong or not to the same coset of $\mathcal{M}_J$ for a certain J with $|J| = 3$. Since the cost to check if two texts belong to the same coset of $\mathcal{M}_J$ for a certain J with $|J| = 3$ requires only 1 XOR operations, the computational cost can be approximated by the cost to construct all the pairs. Moreover, since for each δ_0 and δ_1 the attack needs 2^{13} chosen plaintexts, the operation to construct all the pairs requires $2^{13} \cdot (2^{13} - 1)/2 = 2^{25}$ table look-ups, for a total of $2 \cdot (2^8)^2 \cdot 2^{25} = 2^{42}$ table look-ups.

A way to improve this step is to re-order all the texts using a *merge-sort algorithm*, as we described in details in Algorithm 2. For each δ_0 and δ_1 and for each J with $|J| = 3$, the computational cost to re-order all the elements and to count the collision can be approximated to $2^{13} \cdot (\log 2^{13} + 1) \simeq 2^{17}$ table look-ups, for a total of $4 \cdot 2 \cdot (2^8)^2 \cdot 2^{17} = 2^{36}$ table look-ups, that is $2^{29.7}$ four-round encryption assuming the approximation that one round of AES corresponds to 20 table look-ups. The memory cost is 2^{30} to store all the texts. In this way, the attacker is able to find 2^{32} variants of the whitening key, without working (or finding) any information about the secret S-Box.

For completeness, note that this number can be reduced to 2^8 if one works also on the secret S-Box, as done in [TKKL15].

Practical Verification. The attack just presented has been practically verified: here we report the practical results. For simplicity, here we limit ourselves to report the results when the attacker tries to find two bytes of the secret key - as $k_{0,0} \oplus k_{1,1}$ and $k_{1,2} \oplus k_{2,3}$ - using 2^{13} different chosen plaintexts. These results can be easily extended for the complete attack, as described in the previous text. For completeness, we consider both the two cases in which the attacker (1) re-orders the texts before to count the number of collisions (working only on consecutive ordered elements), or (2) constructs all the possible pairs. This second setting allows to understand better the importance of the re-ordering algorithm in terms of performance/computational cost.

In the second case (that is the one in which all the possible pairs are constructed - no use of the re-ordering algorithm), the expected theoretical computational cost in order to find 2 bytes of the secret key is of $(2^8)^2 \cdot 2^{12} \cdot (2^{13} - 1) = 2^{41} - 2^{28} \simeq 2^{41}$ memory accesses, and it is approximately the same of the practical computational one. Instead, for the first setting (that is the one the re-order the text before to counts the number of collisions) the expected theoretical computational cost in order to find 2 bytes of the secret key is of $4 \cdot (2^8)^2 \cdot 2^{13} \cdot (\log 2^{13} + 1) \simeq 2^{32.8}$ memory accesses. The *average* practical computational cost is approximately of $2^{32.65}$ memory accesses, which is very close to the theoretical one. Finally, 2^{13} chosen plaintexts are (largely) sufficient to find the right value of $k_{0,0} \oplus k_{1,1}$ and $k_{1,2} \oplus k_{2,3}$, as predicted by the theory.

7.3. Impossible Differential Attack on 5-round of AES with a single Secret S-Box

Using the strategy presented in the previous section, it is possible to set up an impossible differential attack on 5 rounds of AES with a secret S-Box. As before, the goal is to find the secret key without needing to discover any information about the S-Box. With respect to previous attacks, the impossible differential attack on 5-round AES has a lower data and computational complexities if one exploits the property that the XOR-sum of three coefficients in the same row of the MixColumns matrix is equal to zero. For simplicity, in the following we also briefly recall the variant of the attack that exploits the property that two coefficients in the same row of the MixColumns matrix are equal.

7.3.1. Idea of the Attack using Equal Coefficients of MC

In the following, we define the set of plaintexts-ciphertexts V_δ with $|V_\delta| = 2^8$:

$$\begin{aligned} V_\delta = &\{(p^i, c^i) \text{ for } i = 0, ..., 2^8 - 1 \,|\, \forall i : p^i_{0,0} \oplus p^i_{1,1} = \delta \quad \text{and} \\ &\text{and} \quad \forall (k, l) \neq \{(0, 0), (1, 1)\} : p^i_{k,l} = p^j_{k,l} \text{ where } i \neq j\}, \end{aligned} \tag{7.2}$$

i.e. plaintexts with 14 constants bytes and with the difference on the other two bytes fixed.

Consider two different pairs (p^1, c^1) and (p^2, c^2) that belong to the same V_δ. By Prop. 13, we know that if $\delta = k_{0,0} \oplus k_{1,1}$, then p^1 and p^2 belong to the same coset of $\mathcal{D}_{0,1,3} \cap \mathcal{C}_0 \subseteq \mathcal{D}_{0,1,3}$ after one round (that is, $R(p^1) \oplus R(p^2) \in \mathcal{D}_{0,1,3} \cap \mathcal{C}_0 \subseteq \mathcal{D}_{0,1,3}$) with probability 1. If $\delta \neq k_{0,0} \oplus k_{1,1}$, they belong to the same coset of $\mathcal{C}_0$ after one round with probability 1, and to the same coset of $\mathcal{D}_{0,1,3} \cap \mathcal{C}_0 \subseteq \mathcal{D}_{0,1,3}$ with probability 2^{-8} (or to the same coset of $\mathcal{D}_J$ for $|J| = 3$ after one round with probability $4 \cdot 2^{-8} = 2^{-6}$).

Consider first the case $\delta = k_{0,0} \oplus k_{1,1}$. Since $R(p^1) \oplus R(p^2) \in \mathcal{D}_{0,1,3}$ for each pair of plaintexts p^1 and p^2 in V_δ, then $R^{(4)} \circ R(p^1) \oplus R^{(4)} \circ R(p^2) = R^{(5)}(p^1) \oplus R^{(5)}(p^2) \notin \mathcal{M}_J$ for $|I| + |J| \leq 4$ with probability 1 due to the 4-round impossible differential distinguisher of Sect. 4.3.3. That is, for each $(p^1, c^1) \neq (p^2, c^2)$

$$Prob\big[R^{(5)}(p^1) \oplus R^{(5)}(p^2) \in \mathcal{M}_J \,|\, (p^1, c^1), (p^2, c^2) \in V_\delta\big] = 0,$$

for each J with $|J| = 1$ and where $\delta := k_{0,0} \oplus k_{1,1}$ is known. As usual, a similar result holds also in the case in which the final MixColumns operation is omitted (in this case, $\mathcal{M}_J$ is replaced by $\mathcal{ID}_J$).

Instead, if $\delta \neq k_{0,0} \oplus k_{1,1}$, note that it's possible that two elements of V_δ belong to the same coset of $\mathcal{M}_J$ for $|J| = 1$ after 5-round. In particular, the probability that two elements p and q in V_δ belong to the same coset of $\mathcal{M}_J$ after 5-round for a certain J with $|J| = 1$ is approximately[6] $4 \cdot 2^{-96} = 2^{-94}$.

The idea is to exploit these different probabilities in order to find the key. In particular, a key candidate δ can be declared wrong if there is at least one collision, i.e. two different pairs of texts (p^1, c^1) and (p^2, c^2) such that $p^1 \oplus p^2 \in V_\delta$ and $c^1 \oplus c^2 \in \mathcal{M}_J$ for $|J| = 1$. Thus, in the following we look for the minimum number of texts necessary to have at least one collision *for each* $\delta \neq k_{0,0} \oplus k_{1,1}$ with high probability.

[6] The exact probability for a *wrong* $\delta \neq k_{0,0} \oplus k_{1,1}$ is given by $Pr(R^{(5)}(p^1) \oplus R^{(5)}(p^2) \in \mathcal{M}_J \,|\, p^1 \oplus p^2 \in V_\delta) = 2^{-6} \cdot 0 + (1 - 2^{-6}) \cdot 4 \cdot 2^{-96} = 2^{-94} - 2^{-100} \simeq 2^{-94}$, which is derived considering the two cases $R(p^1) \oplus R(p^2) \in \mathcal{D}_J$ and $R(p^1) \oplus R(p^2) \notin \mathcal{D}_J$ for $|J| = 3$.

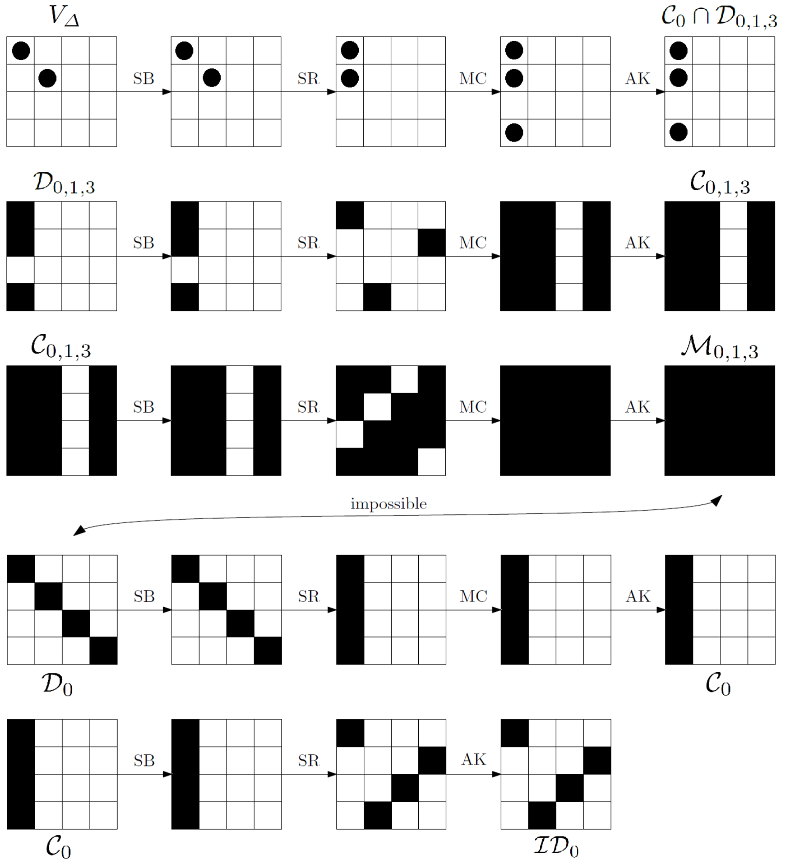


Figure 7.4.: *5-Round Secret Key Distinguisher for AES with a single secret S-Box* from [GRR16] – data complexity $2^{98.2}$. The choice of the plaintexts (i.e. $p_{0,0} \oplus p_{1,1} = k_{0,0} \oplus k_{1,1}$) guarantees that after one round there are only three bytes with non-zero difference instead of four, that is the plaintexts belong to the same coset of $\mathcal{C}_0 \cap \mathcal{D}_{0,1,3}$. The probability the two ciphertexts belong to the same coset of $\mathcal{M}_k$ for $|k| = 1$ is zero. White box denotes denotes a byte with a zero-difference, while a black box denotes a byte with non-zero difference.

Before to proceed, note that a similar impossible differential attack can be set up for 4-round AES with secret S-Box, exploiting the fact that two elements in the same coset of $\mathcal{D}_J$ can not belong to the same coset of $\mathcal{C}_I$ after three rounds for $|I| + |J| \leq 4$.

7.3.2. Attack using Zero XOR-sum of some Coefficients of MC

Here we show how to set up an impossible differential attack on 5-round AES [Gra18a] that exploits the fact that a sum of coefficients of the MixColumns matrix is equal to zero (e.g. (7.1)), and improves the one just recalled [GRR16].

For a fixed $a \in \mathcal{D}_0^\perp$ (i.e. $a_{i,i} = 0$ for $i = 1, 2, 3$), consider a set of plaintexts of the form:

$$V_\delta \equiv \left\{ a \oplus \begin{bmatrix} x & 0 & 0 & 0 \\ 0 & x \oplus \delta_{1,1} & 0 & 0 \\ 0 & 0 & x \oplus \delta_{2,2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \middle| \, \forall x \in \mathbb{F}_{2^8} \right\} \tag{7.3}$$

and let $\delta \equiv (\delta_{1,1}, \delta_{2,2})$. Since

$$M_{r,1} \oplus M_{r,2} \oplus M_{r,3} = 0 \qquad \text{for} \qquad r = 0, 1,$$

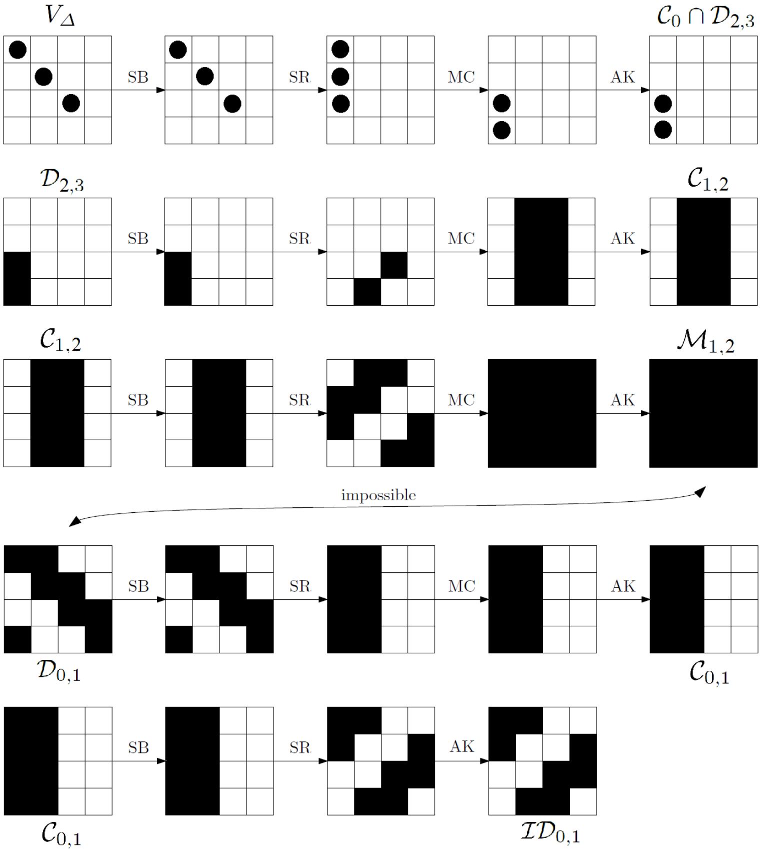


Figure 7.5.: *5-Round secret-key distinguisher for AES with a single secret S-Box* from [Gra18a] – data complexity $2^{76.4}$. The choice of the plaintexts (i.e. $p_{0,0} \oplus p_{i,i} = k_{0,0} \oplus k_{i,i}$ for $i = 1, 2$) guarantees that after one round there are only two bytes with non-zero difference instead of four, that is the plaintexts belong to the same coset of $\mathcal{C}_0 \cap \mathcal{D}_{2,3}$. Thus, the probability the two ciphertexts belong to the same coset of $\mathcal{M}_K$ for $|K| = 2$ is zero. White box denotes denotes a byte with a zero-difference, while a black box denotes a byte with non-zero difference.

it follows by Prop. 14 that the set V_δ is mapped into a coset of $\mathcal{C}_0 \cap \mathcal{D}_{2,3}$ with probability 1 after one round if $\delta_{1,1} = k_{1,1} \oplus k_{0,0}$ *and* $\delta_{2,2} = k_{2,2} \oplus k_{0,0}$. In the other cases, that is if $\delta_{1,1} \neq k_{1,1} \oplus k_{0,0}$ and/or $\delta_{2,2} \neq k_{2,2} \oplus k_{0,0}$ the set V_δ is mapped into a coset of $\mathcal{C}_0$ with probability 1, and into a coset of $\mathcal{C}_0 \cap \mathcal{D}_I \subseteq \mathcal{D}_I$ for a certain I with $|I| = 2$ with probability $6 \cdot 2^{-16} = 3 \cdot 2^{-15}$.

Since $Prob\big[R^4(x) \oplus R^4(y) \in \mathcal{M}_J \,|\, x \oplus y \in \mathcal{D}_I\big] = 0$ for $|I| + |J| \leq 4$ - see (4.6), if $\delta_{1,1} = k_{1,1} \oplus k_{0,0}$ *and* $\delta_{2,2} = k_{2,2} \oplus k_{0,0}$, it follows that given two plaintexts in the same coset of V_δ, then the corresponding ciphertexts after five rounds can not belong to the same coset of $\mathcal{M}_J$ for $|J| = 2$:

$$Prob\big[R^5(x) \oplus R^5(y) \in \mathcal{M}_J \,|\, x, y \in V_\delta \quad \text{and} \quad \delta_{i,i} = k_{i,i} \oplus k_{0,0} \text{ for } i = 1, 2\big] = 0.$$

In the other cases - if $\delta_{1,1} \neq k_{1,1} \oplus k_{0,0}$ and/or $\delta_{2,2} \neq k_{2,2} \oplus k_{0,0}$, given two plaintexts in the same coset of V_δ, then the corresponding ciphertexts after 5-round belong to the same coset of $\mathcal{M}_J$ for $|J| = 2$ with prob. $6 \cdot 2^{-64} = 3 \cdot 2^{-63}$. The idea is to exploit this difference in the probabilities to recover the secret key.

Comparison with the previous Impossible-Differential Attack. For completeness, we briefly discuss the difference with the attack proposed in [GRR16] and illustrated in Fig. 7.4. In this last case, a similar set V_δ is defined, and the idea is to exploit the fact two elements of each row of the MixColumns matrix are equal. As before, for the right guessed key and given two plaintexts in

```
Data: $2^{74.4}$ different sets $V_\delta$ defined as in (7.3) - $2^{58.4}$ for each $\delta \equiv (\delta_{1,1}, \delta_{2,2})$ - and
      corresponding ciphertexts after 5 rounds
Result: $k_{0,0} \oplus k_{1,1}$ and $k_{0,0} \oplus k_{2,2}$
for each $\delta_{1,1}$ from 0 to $2^8 - 1$ and each $\delta_{2,2}$ from 0 to $2^8 - 1$ do
    $flag \leftarrow 0$;
    for each set $V_\delta$ do
        for each $I \subseteq \{0,1,2,3\}$ with $|I| = 2$ do
            let $(p^i, c^i)$ for $0 \leq i \leq 2^8 - 1$ be the $2^8$ (plaintexts, ciphertexts) of $V_\delta$;
            re-order this set of elements w.r.t. the partial order $\preceq$ defined in analogous way of
             Def. 14 s.t. $c^i \preceq c^{i+1}$ $\forall i$;                          // $\preceq$ depends on $I$
            for $i$ from 0 to $2^8 - 2$ do
                if $c^i \oplus c^{i+1} \in \mathcal{M}_I$ then
                    $flag \leftarrow 1$;
                    next $\delta$;
                end
            end
        end
    end
    if $flag = 0$ then
        identify $\delta_{1,1}$ as candidate for $k_{0,0} \oplus k_{1,1}$ and $\delta_{2,2}$ as candidate for $k_{0,0} \oplus k_{2,2}$;
    end
end
return Candidates for $k_{0,0} \oplus k_{1,1}$ and $k_{0,0} \oplus k_{2,2}$.    // Only one candidate with Prob.
 95%
```

Algorithm 7: *Impossible Differential Attack on 5 rounds of AES with a single secret S-Box.* For simplicity, the goal of the attack is to find two bytes of the key - $k_{0,0} \oplus k_{1,1}$ and $k_{0,0} \oplus k_{2,2}$. The same attack on the other diagonals can be used to recover the entire key up to 2^{32} variants.

the same coset of V_δ, then the corresponding ciphertexts after 5-round can not belong to the same coset of $\mathcal{M}_J$ for $|J| = 1$ The main difference regards the case of a wrong guessed key, for which the previous event happens with prob. 2^{-94}. As a result, one needs more texts to detect the wrong guessed keys.

7.3.3. Data Complexity and Computational Cost

Data Cost. First of all, consider the attack on 2 bytes of the secret key. In order to discard a wrong candidate δ of the key, it is sufficient that at least one set V_δ for which a pair of ciphertexts belong to the same coset of $\mathcal{M}_J$ with $|J| = 2$ exists (note that this can never happen for the right value of δ - the secret key). Since there are $2^{16} - 1$ wrong candidates, in order to have a total probability of success of 95%, such a set must exist for each δ with probability higher than $(0.95)^{2^{-16}} \simeq 99.999922\%$.

Given a set V_δ, it is possible to construct approximately $2^7 \cdot (2^8 - 1) = 2^{15}$ different pairs of ciphertexts. Since each pair can belong to the same coset of $\mathcal{M}_J$ with a probability of $3 \cdot 2^{-63}$, given n different pairs, the probability that at least one of them belong to the same coset of $\mathcal{M}_J$ is $1 - (1 - 3 \cdot 2^{-63})^n$. By simple computation, the condition $1 - (1 - 3 \cdot 2^{-63})^n > 0.99999922$ is satisfied for $n > 2^{65.23}$. Since each set V_δ is composed of 2^{15} pairs and since one has to repeat the attack for each possible value of δ, the attacker needs approximately $2^{65.23} \cdot 2^{-7} \cdot 2^{16} = 2^{74.23}$ chosen plaintexts to find two bytes of the secret key (note that each set V_δ contains 2^8 texts, so $2^{-15} \cdot 2^8 = 2^{-7}$).

The idea is to repeat this attack 4 times in order to find 8 bytes of the key (i.e. 2 for column). In this case, for each candidate δ of the key at least one set V_δ with the previous property must exist with probability higher $(0.95)^{2^{-18}} \simeq 99.99998\%$. Using the same calculation as before, one needs approximately $n > 2^{65.37}$ pairs of ciphertexts for each δ, i.e. approximately $2^{50.37}$ different sets V_δ.

Finally, in order to find the final 4 bytes of the key (remember that we are to find it up to 2^{32} variants), the idea is to repeat again the previous attack. However, note that in this case the attacker must guess only one byte of the key for each diagonal instead of two (since two of three differences are already known). Thus, for each wrong δ, at least one set for which two ciphertexts belong to the same coset of $\mathcal{M}_J$ with $|J| = 2$ must exist with probability higher $(0.95)^{2^{-10}} \simeq 99.995\%$. Using the same calculation as before, one needs approximately $n > 2^{64.73}$ pairs of ciphertexts for each δ, that is approximately $2^{57.73}$ different sets V_δ. It follows that the total data complexity is approximately of $4 \cdot 2^{58.37} \cdot 2^{16} + 4 \cdot 2^{57.73} \cdot 2^{8} = 2^{76.374}$ chosen plaintexts.

Computational Cost. Using the re-ordering algorithm proposed in Algorithm 7, the computational cost is well approximated by $4 \cdot 4 \cdot 2^{58.37} \cdot 2^{16} \cdot (\log 2^8 + 1) = 2^{81.54}$ table look-ups, or approximately $2^{74.9}$ five-round encryptions (20 table look-ups $\approx$ 1-round of encryption). For comparison, the attack previously proposed requires 2^{102} chosen plaintexts and computational cost is of $2^{100.4}$ five-round encryptions.

7.4. Multiple-of-n Attack on 5-round AES with a secret S-Box

Another possible way to attack round-reduced AES with a single secret S-Box is to exploit the multiple-of-n property. As for the impossible differential attack just presented, we consider separately the cases in which *(1st)* two coefficients of the MixColumns matrix are equal and *(2nd)* the XOR-sum of some coefficients in the same row of the MixColumns matrix is equal to zero.

7.4.1. Attack using Equal Coefficients of MC

The idea is choose a particular set of plaintexts $\mathcal{A}_\delta$ (which depends on a variable δ), such that only for a particular value of δ - which depends on the secret key - the number of collisions among the ciphertexts in the same coset of $\mathcal{M}_I$ with $|I| = 3$ after 5 rounds is a multiple of 2 (i.e. it is an even number) with probability 1. Since for all the other values of δ this event happens only with probability 1/2, it is possible to discover the right key. Thus, for a fixed $a \in \mathcal{D}_1^\perp$ (i.e. $a_{0,1} = a_{1,2} = 0$), let $\mathcal{A}_\delta$ be the set of plaintexts of the form:

$$\mathcal{A}_\delta \equiv \left\{ a \oplus \begin{bmatrix} y_0 & x & 0 & 0 \\ 0 & y_1 & x \oplus \delta & 0 \\ 0 & 0 & y_2 & 0 \\ 0 & 0 & 0 & y_3 \end{bmatrix} \middle| \forall x, y_0, ..., y_3 \in \mathbb{F}_{2^8} \right\}. \tag{7.4}$$

Given a set $\mathcal{A}_\delta$, we claim that if $\delta = k_{0,1} \oplus k_{1,2}$ then the number of collisions after 5 rounds in the same coset of $\mathcal{M}_I$ for a fixed $I \subseteq \{0,1,2,3\}$ with $|I| = 3$ is a multiple of 2 with probability 1.

Proposition 16 ([Gra18a]). *Consider a set of plaintexts $\mathcal{A}_\delta$ defined as in (7.4), and the corresponding ciphertexts after 5 rounds. If $\delta = k_{0,1} \oplus k_{1,2}$, then the number of different pairs of ciphertexts that belong to the same coset of $\mathcal{M}_I$ for a fixed $I \subseteq \{0,1,2,3\}$ with $|I| = 3$ is a multiple of 2.*

Proof. Let $\delta = k_{0,1} \oplus k_{1,2}$. After one round, there exists b such that the set $\mathcal{A}_\delta$ is mapped into

$$R(\mathcal{A}_\delta) \equiv \left\{ b \oplus \begin{bmatrix} z_0 & w & 0 & 0 \\ z_1 & 0x03 \cdot w & 0 & 0 \\ z_2 & 0 & 0 & 0 \\ z_3 & 0x02 \cdot w & 0 & 0 \end{bmatrix} \middle| \forall w, z_0, ..., z_3 \in \mathbb{F}_{2^8} \right\}.$$

Consider two elements $z, z' \in R(\mathcal{A}_\delta)$ generated respectively by $z \equiv (z_0, z_1, z_2, z_3, w)$ and $z' \equiv (z'_0, z'_1, z'_2, z'_3, w)$, and consider separately the two cases $z_1 \neq z'_1$ and $z_1 = z'_1$. The idea is to show that in the first case (i.e. the set of all the different pairs of elements for which the condition $z_{1,1} \neq z'_{1,1}$ holds) the number of collisions is a multiple of 2, while in the second case (i.e. the set of all the different pairs of elements for which the condition $z_1 = z'_{1,1}$ holds) the number of collisions is a multiple of 256. In particular, consider two elements $z, z' \in R(\mathcal{A}_\delta)$ generated respectively by $z \equiv (z_0, z_1, z_2, z_3, w)$ and $z' \equiv (z'_0, z'_1, z'_2, z'_3, w)$ with $z_1 \neq z'_1$. For a fixed $I \in \{0, 1, 2, 3\}$ with $|I| = 3$, the idea is to show that $R^4(z) \oplus R^4(z') \in \mathcal{M}_I$ *if and only if* $R^4(v) \oplus R^4(v') \in \mathcal{M}_I$ where the texts $v, v' \in R(\mathcal{A}_\delta)$ are generated respectively by $v \equiv (z_0, z'_1, z_2, z_3, w)$ and $v' \equiv (z'_0, z_1, z'_2, z'_3, w)$. Similarly, consider the case $z_1 = z'_1$. For this case, the idea is to prove that $z, z' \in R(\mathcal{A}_\delta)$ satisfy the condition $R^4(z) \oplus R^4(z') \in \mathcal{M}_I$ if and only if each pair of elements $v, v' \in R(\mathcal{A}_\delta)$ generated respectively by $v \equiv (z_0, v_1, z_2, z_3, w)$ and $v' \equiv (z'_0, v_1, z'_2, z'_3, w)$ for each $v_1 \in \mathbb{F}_{2^8}$ have the same property, that is $R^4(v) \oplus R^4(v') \in \mathcal{M}_I$. Since there are $2^8 = 256$ different values for v_1, then the number of collisions must be a multiple of 256. It follows that there exist $n', n'' \in \mathbb{N}$ such that the total number of collisions n can be written as $n = 2 \cdot n' + 256 \cdot n'' = 2 \cdot (n' + 128 \cdot n'')$. In other words, the total number of collisions is a multiple of 2.

More details of the proof can be found in [Gra17a, App. E]. □

Consider now the case $\delta \neq k_{0,1} \oplus k_{1,2}$. In this case, the previous proposition does not hold and the number of collisions is a multiple of 2 only with probability 1/2. Indeed, let $\delta \neq k_{0,1} \oplus k_{1,2}$. By simple computation, there exists a constant b such that the set $\mathcal{A}_\delta$ is mapped after one round into

$$R(\mathcal{A}_\delta) \equiv b \oplus \begin{bmatrix} z_{0,0} & \text{0x02} \cdot \text{S-Box}(x \oplus k_{0,1}) \oplus \text{0x03} \cdot \text{S-Box}(x \oplus \delta \oplus k_{1,1}) & 0 & 0 \\ z_{1,1} & \text{S-Box}(x \oplus k_{0,1}) \oplus \text{0x02} \cdot \text{S-Box}(x \oplus \delta \oplus k_{1,1}) & 0 & 0 \\ z_{2,2} & \text{S-Box}(x \oplus k_{0,1}) \oplus \text{S-Box}(x \oplus \delta \oplus k_{1,1}) & 0 & 0 \\ z_{3,3} & \text{0x03} \cdot \text{S-Box}(x \oplus k_{0,1}) \oplus \text{S-Box}(x \oplus \delta \oplus k_{1,1}) & 0 & 0 \end{bmatrix}$$

for each x and for each $z_{0,0}, ..., z_{3,3}$. Note that this is a subset (*not* a subspace) of a coset of $\mathcal{C}_{0,1}$. Thus, assume that two elements $z, z' \in R(\mathcal{A}_\delta)$ belong to the same coset of $\mathcal{M}_I$ after 4 rounds. Since the second column of $R(\mathcal{A}_\delta)$ can take only a limited number of values, working in the same way as before it is not possible to guarantee that other pairs of elements - defined by a different combinations of the variables - have the same property with prob. 1. It follows that in this case the number of collisions is a multiple of 2 only with probability 1/2 (this result has been practically verified).

Note that each set contains 2^{40} different texts, that is approximately $2^{39} \cdot (2^{40} - 1) \simeq 2^{79}$ different pairs of ciphertexts. Since the probability that two ciphertexts belong to the same coset of $\mathcal{M}_I$ for $|I| = 3$ is 2^{-32}, the number of collisions is approximately $2^{79} \cdot 2^{-32} = 2^{47}$. We emphasize that for the right key this number is exactly a multiple of 2 with probability 1, while for wrong guessed keys this happens only with probability 1/2. Using these considerations, it is possible to find the right key up to 2^{32} variants.

Data and Computational Costs

Data Cost. To compute the data cost, we first analyze the case in which the goal is to discover only one byte (in particular, the difference of two bytes) of the right key with probability greater than 95%. A candidate value of δ can be claimed to be wrong if there exists at least a set $\mathcal{A}_\delta$ for which the number of collisions after five rounds is an odd number. Since there are only $2^8 - 1$ different possible values for δ, one needs that such a set $\mathcal{A}_\delta$ exists with probability higher than $(0.95)^{1/255} = 99.98\%$ (since the tests for different δ are independent, the total probability of success is higher than $0.9998^{256} = 0.95$).

Since the probability that the number of collisions for a given set $\mathcal{A}_\delta$ is odd is 50%, 4 different sets $\mathcal{A}_\delta$ (note that one can count the number of collisions in $\mathcal{M}_I$ for all the 4 different I with $|I| = 3$, for a total of 16 possible tests) are sufficient to deduce the right δ with probability higher than 95%,

```
Data: $2^{10}$ different sets $\mathcal{A}_\delta$ defined as in (7.4) - 4 different sets for each $\delta$ - and
      corresponding ciphertexts after 5 rounds
Result: $k_{0,0} \oplus k_{1,1}$
for each $\delta$ from 0 to $2^8 - 1$ do
    $flag \leftarrow 0$;
    for each set $\mathcal{A}_\delta$ do
        let $(p^i, c^i)$ for $i = 0, ..., 2^{40} - 1$ be the $2^{40}$ (plaintexts, ciphertexts) of $\mathcal{A}_\delta$;
        for all $j \in \{0, 1, 2, 3\}$ do
            Let $W[0, ..., 2^{32} - 1]$ be an array initialized to zero;
            for $i$ from 0 to $2^{40} - 1$ do
                $x \leftarrow \sum_{k=0}^{3} MC^{-1}(c^i)_{k,j-k} \cdot 256^k$; // $MC^{-1}(c^i)_{k,j-k}$ denotes the byte of
                  $MC^{-1}(c^i)$ in row $k$ and column $j-k \mod 4$ $W[x] \leftarrow W[x] + 1$; // $W[x]$
                  denotes the value stored in the $x$-th address of the array $W$
            end
            $n \leftarrow \sum_{i=0}^{2^{32}-1} W[i] \cdot (W[i] - 1)/2$;
            if $(n \mod 2) \neq 0$ then
                $flag \leftarrow 1$ (next $\delta$);
            end
        end
    end
    if $flag = 0$ then
        identify $\delta$ as candidate for $k_{0,0} \oplus k_{1,1}$;
    end
end
return Candidates for $k_{0,0} \oplus k_{1,1}$.               // Only one candidate with Prob. 95%
```

Algorithm 8: *Key-Recovery Attack on 5 rounds of AES with a single secret S-Box.* For simplicity, the goal of the attack is to find one byte of the key - $k_{0,0} \oplus k_{1,1}$. The same attack is used to recover the entire key up to 2^{32} variants.

since $2^{-16} \leq 1 - 0.9998 = 2^{-12.3}$. It follows that the cost to find 1 byte of the key is of 4 (cosets) $\cdot 2^{40}$ (number of texts in $\mathcal{A}_\delta$) $\cdot 2^8$ (values of δ) $= 2^{50}$ chosen plaintexts.

In order to find the entire key up to 2^{32} possible variants, the idea is to repeat the attack 12 times, i.e. 3 times for each column. By analogous calculation[7], it follows that 16 tests (that is 4 different sets $\mathcal{A}_\delta$ - note that there are four different I with $|I| = 3$) are sufficient to deduce the right δ with total probability higher than 95%. Thus, the data cost of the attack is of $12 \cdot 2^{50} = 2^{53.6}$ chosen plaintexts.

Computational Cost. In order to count the number of collisions, one can exploit *data structure* - the complete pseudo-code of such an algorithm is given in Algorithm 8. This method allows to minimize the computational cost, which is well approximated by $2^{55.6}$ table look-ups or approximately $2^{48.96}$ five-rounds encryptions (20 table look-ups $\approx$ 1 round of encryption).

Practical Verification Using a C/C++ implementation[8], we have practically verified the attack just described on a small-scale variant of AES, as presented in [CMR05] - not on real AES due to the large computational cost of the attack. We emphasize that Prop. 16 is independent of the fact that each word is composed of 8 or 4 bits. Thus, our verification on the small-scale variant of AES is

[7] In this case, one needs that for each one of the $2^8 - 1$ wrong possible values for δ, at least one set $\mathcal{A}_\delta$ for which the number of collision is odd exists with probability higher than $(0.9998)^{1/12} = 99.99835\%$.

[8] The source codes of the attacks on AES with a secret S-Box in this section are available at `https://github.com/Krypto-iaik/Attacks_AES_SecretSBox2`

strong evidence for it to hold for the real AES. The main differences between this small-scale AES and the real AES regard the total computational cost.

For simplicity, we limit ourselves here to report the result for an attack on a single byte of the key, e.g. $k_{0,0} \oplus k_{1,1}$. For small-scale AES, since there are only $2^4 - 1$ possible candidates, it is sufficient that for each wrong candidate of $k_{0,0} \oplus k_{1,1}$ a set $\mathcal{A}_\delta$ for which the number of collisions is odd exists with probability $(0.95)^{2^{-4}} = 99.659\%$. It follows that 9 tests (that is 3 different sets $\mathcal{A}_\delta$) for each candidate of $k_{0,0} \oplus k_{1,1}$ are sufficient to find the right value. Using the same procedure just presented based on data-structure, the theoretical computational cost is well approximated by $4 \cdot 3 \cdot 2^4 \cdot (2^{20} + 2 \cdot 2^{16}) \simeq 2^{27.75}$ table look-ups.

Our tests confirm that 3 different sets $\mathcal{A}_\delta$ are largely sufficient to find the key. The average practical computational cost is of $2^{26.3}$ table look-ups using a data-structure. To explain the (small) difference with the theoretical value, note that the theoretical value is computed in the worst case. As example, when a candidate of the key is found to be wrong, it is not necessary to complete the verification for all the other sets $\mathcal{A}_\delta$ or indexes I, but it is sufficient to discard it and to test the next candidate.

7.4.2. The Attack using Zero XOR-Sum of some Coefficients of MC

Here we show how to adapt the previous attack in order to exploits the property that the sum of three coefficients of each row of the MixColumns matrix M is equal to zero.

For a fixed a, consider a set of plaintexts $\mathcal{A}''_\delta$ which depends on the guessed value of the key δ of the form:

$$\mathcal{A}''_\delta \equiv \left\{ a \oplus \begin{bmatrix} 0 & y & 0 & 0 \\ 0 & x & y \oplus \delta_{1,2} & 0 \\ 0 & 0 & x \oplus \delta_{2,2} & y \oplus \delta_{2,3} \\ 0 & 0 & 0 & x \oplus \delta_{3,3} \end{bmatrix} \middle| \forall x, y \in \mathbb{F}_{2^8} \right\} \tag{7.5}$$

where $\delta \equiv (\delta_{1,2}, \delta_{2,2}, \delta_{2,3}, \delta_{3,3})$. Given a set $\mathcal{A}''_\delta$, we claim that the number of collisions among the ciphertexts in the same coset of $\mathcal{M}_I$ for a fixed $I \subseteq \{0, 1, 2, 3\}$ with $|I| = 3$ after 5 rounds is a multiple of 2. More formally:

Proposition 17 ([Gra18a]). *Consider a set of plaintexts $\mathcal{A}''_\delta$ defined as in (7.5), and the corresponding ciphertexts after 5 rounds. If $\delta_{i,i} = k_{1,1} \oplus k_{i,i}$ and $\delta_{j,j+1} = k_{0,1} \oplus k_{j,j+1}$ for $i = 2, 3$ and $j = 1, 2$ (the indexes are taken modulo 4), then the number of different pairs of ciphertexts that belong to the same coset of $\mathcal{M}_I$ for a fixed $I \subseteq \{0, 1, 2, 3\}$ with $|I| = 3$ is a multiple of 2.*

Proof. Let $\delta_{i,i} = k_{i,i} \oplus k_{1,1}$ for $i = 2, 3$ and $\delta_{j,j+1} = k_{j,j+1} \oplus k_{0,1}$ for $j = 1, 2$. By simple computation, there exists a constant b such that a set $\mathcal{A}''_\delta$ is mapped after one round into

$$R(\mathcal{A}''_\delta) \equiv \left\{ b \oplus \begin{bmatrix} \texttt{0x03} \cdot z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \texttt{0x02} \cdot w & 0 & 0 \\ \texttt{0x02} \cdot z & \texttt{0x03} \cdot w & 0 & 0 \end{bmatrix} \middle| \forall z, w \in \mathbb{F}_{2^8} \right\}.$$

Consider a pair of texts $t^1, t^2 \in R(\mathcal{A}''_\delta)$ generated respectively by $t^1 = (z, w)$ and $t^2 = (z', w')$. The idea is to consider the following two cases separately: (1) $z = z'$ and $w \neq w'$ (or vice-versa) and (2) $z \neq z'$ and $w \neq w'$, and to show that in the first case (1) the number of collisions is a multiple of 256, while in the second case (2) the number of collisions is a multiple of 2. In particular, consider a pair of texts $t^1, t^2 \in R(\mathcal{A}''_\delta)$ generated respectively by $t^1 = (z, w)$ and $t^2 = (z', w')$ with $z \neq z'$ and $w \neq w'$. The idea is to show that $R^4(t^1) \oplus R^4(t^2) \in \mathcal{M}_I$ *if and only if* $R^4(s^1) \oplus R^4(s^2) \in \mathcal{M}_I$ for $|I| = 3$, where the texts $s^1, s^2 \in R(\mathcal{A}''_\delta)$ are generated respectively by $s^1 = (z, w')$ and $s^2 = (z', w)$. Similarly, consider the case $z \neq z'$ and $w = w'$ (or vice-versa). As before, the idea is to prove that $t^1, t^2 \in R(\mathcal{A}''_\delta)$ satisfy the condition $R^4(t^1) \oplus R^4(t^2) \in \mathcal{M}_I$ for $|I| = 3$ if and only if all the pairs of texts $s^1, s^2 \in R(\mathcal{A}''_\delta)$ generated respectively by $t^1 = (z, s)$ and $t^2 = (z', s)$ for all $s \in \mathbb{F}_{2^8}$ have the same property. Thus, there exist $n', n'' \in \mathbb{N}$ such that the total number of collisions n can be written as $n = 2 \cdot n' + 256 \cdot n'' = 2 \cdot (n' + 128 \cdot n'')$, that is n is a multiple of 2.

More details of the proof can be found in [Gra17a, App. G]. □

While for $\delta_{i,i} = k_{i,i} \oplus k_{1,1}$ for $i = 2, 3$ and $\delta_{j,j+1} = k_{j,j+1} \oplus k_{0,1}$ for $j = 1, 2$ it is possible to guarantee that the total number of collisions is a multiple of 2 with probability 1, no analogous result holds for the other cases. That is, if $\delta_{i,i} \neq k_{i,i} \oplus k_{1,1}$ for $i = 2, 3$ or/and $\delta_{j,j+1} \neq k_{j,j+1} \oplus k_{0,1}$ for $j = 1, 2$, then the total number of collisions is a multiple of 2 with probability 50%.

Data and Computational Costs. Since the procedure of the attack is completely equivalent to the one just described, we limit ourselves here to report the data and computational costs of the attack. The total data complexity is approximately of $2 \cdot 2^{52.248} + 12 \cdot 2^{16} \cdot 2^{16} = 2^{53.25}$ chosen plaintexts, while - using the re-ordering algorithm proposed in Algorithm 2 - the computational cost is well approximated by $2 \cdot 4 \cdot 19 \cdot 2^{32} \cdot 2^{16} \cdot (\log 2^{16} + 1) \simeq 2^{59.25}$ table look-ups, or approximately $2^{52.6}$ five-round encryptions.

Practical Verification Using a C/C++ implementation, we have practically verified the attack just described on a small-scale variant of AES [CMR05] - not on real AES due to the large computational cost of the attack. As before, we emphasize that Prop. 17 is independent of the fact that each word is composed of 8 or 4 bits and that our verification on the small-scale variant of AES is strong evidence for it to hold for the real AES.

For simplicity, we limit ourselves here to report the result for the attack on four bytes of the key, e.g. $k_{2,2} \oplus k_{1,1}$, $k_{3,3} \oplus k_{1,1}$, $k_{0,1} \oplus k_{1,2}$ and $k_{0,1} \oplus k_{2,3}$. For small-scale AES, since there are $(2^4)^4 = 2^{16}$ candidates for the four bytes of the key, it is sufficient that a set $\mathcal{A}''_\delta$ for which the number of collisions is odd exists for each wrong candidate with probability higher than $(0.95)^{2^{-16}}$. Thus, $22 \cdot 2 = 44$ tests (i.e. 11 different sets $\mathcal{A}_\delta$) for each candidate δ are sufficient to find the right value. Re-ordering the texts as described previously, the theoretical computational cost is well approximated by $11 \cdot 2^{16} \cdot 4 \cdot 2^8 \cdot (\log 2^8 + 1) \simeq 2^{32.6}$ table look-ups.

Our tests confirm that 2 different sets $\mathcal{A}_\delta$ are largely sufficient to find the key. The average practical computational cost is of $2^{29.7}$ table look-ups. As before, the difference is explained by the fact that in general it is possible to discard wrong candidates without considering all the corresponding 11 sets $\mathcal{A}''_\delta$.

8
Open-Key Distinguishers for AES

The concept of known-key distinguishers was introduced by Knudsen and Rijmen in [KR07]. Informally, in the classical single secret-key setting, the attacker does not know the randomly generated key and aims to recover it or builds a (secret-key) distinguisher that allows to distinguish the cipher from a random permutation. The security model in known-key attacks is quite different though: the attacker knows the randomly drawn key the block cipher operates with, and aims to find a structural property for the cipher under the known key - a property which an ideal cipher (a permutation drawn at random) would not have. A more relaxed version - called chosen-key distinguisher - can also be considered, where the adversary is assumed to have a full control over the key. This model was introduced in [BKN09], and has been extended to a related-key attack on the full-round AES-256. In the literature, these two models of security of block ciphers are usually denoted as *open key model*, where the adversary knows or even chooses keys.

Since their introductions, open-key attacks have been a major research topic in the symmetric-key community. To provide some examples, besides AES (discussed in the following) known-key distinguishers have been proposed for full PRESENT [BPW15] – one of the most studied lightweight block cipher proposed at CHES 2007 – and for Feistel networks [SY11]. This is justified by the fact that, even if open-key distinguishers could be considered less relevant than secret-key ones, they anyway allow to learn something about the security margin of a cipher. For example, if it is not possible to find distinguishers for a block cipher when the key is given, then one cannot find a distinguisher when the key is secret. Secondly and more important, block ciphers and hash functions are very close cryptographic primitives, as the latter can be built from the former and vice versa. For example, the Davies-Meyer construction, the Miyaguchi-Preneel construction or the Sponge construction can transform a secure block cipher[1] into a secure compression function. In a hash setting, block cipher security models such as the known-key model (or the chosen-key model) make sense since in practice the attacker has full access and control over the internal computations. Moreover, an attack in these models depicts a structural flaw of the cipher, while it should be desired to work with a primitive that does not have any flaw, even in the most generous security model for the attacker. A classical example is the devastating effect on the compression function security of weak keys for a block cipher [WPS+12], which are usually considered as a minor flaw for a block cipher if the set of these weak-keys is small. Therefore, the security notions to consider for a block cipher will vary depending if this block cipher will be used in a hash function setting or not.

Citing Knudsen and Rijmen [KR07], "*imagine a block cipher*" for which a known-key distinguisher exists, "*but where no efficient attacks are known in the traditional black-box model. Should we recommend the use of such a cipher? We do not think so!*"

Any Concrete Implementable Cipher can be Trivially Distinguished From an Ideal Cipher. The open-key model received scrutiny from a more theoretical side too. Traditionally, block ciphers have been examined under the classical notion of indistinguishability. In that setting a block cipher $E(\cdot)$ is claimed secure if it is (computationally) indistinguishable from a fixed random permutation $\Pi(\cdot)$ with the same domain and range as $E(\cdot)$. In other words, an attacker

[1] Actually, the Sponge construction can turn a pseudo-random permutation into a secure compress function. Such pseudo-random permutation can be obtained by a secure cipher by fixing the key.

has to distinguish between $E(\cdot)$ and $\Pi(\cdot)$ when placed in either real or ideal worlds, respectively. Indistinguishability has been established as the *de facto* security notion for block ciphers because in the encryption setting the intended use of the cipher key is in a secret manner.

On the other hand, *indistinguishability cannot provide strong security guarantees in the open key model*: as it was shown already in [CGH04], any concrete implementable cipher (like the AES instantiated by a *known* key) can be trivially distinguished from an ideal cipher. For instance, consider the following straightforward distinguishability attack. Assume *the goal is to distinguish if an oracle is instantiated by a cipher $E_K(\cdot)$ or by an ideal cipher $\Pi(K, \cdot)$ under a known/chosen key K.* Given a query X, one gets Y (which can be $Y = E_K(\cdot)$ or $Y = \Pi(K, X)$). Since the details of $E_K(\cdot)$ and the key K are known, one can simply compute $Y' = E_K(X)$. If $Y' = Y$, one can conclude that the oracle is instantiated by $E_K(\cdot)$. We emphasize that *the "weak point" of such an indistinguishability notion is that it allows access to the internal primitives $E_K(\cdot)$.*

Despite this cumulative impact in the symmetric-key community over the last years, open-key distinguishers/attacks have been known to be difficult to formalize since, formally speaking, it is not clear what an exploitable structural property of a block cipher under a known key is. There have been several attempts to solve the problem in general but we are not aware of any published result here.

8.1. "*Weak*" Known-Key Distinguisher

As we have just seen, *every block cipher with a fixed underlying primitive is vulnerable to a known-key distinguisher*. To overcome this problem and since our work is more practically oriented, in the following we limit ourselves to work in the "weak cipher model (WCM)", as introduced in [MP15]. Roughly speaking, in this model we just consider distinguishers that exploit *properties which have no connection with the details of the underlying primitive E and that are independent of the (value of the) key.*

The Weak Cipher Model (WCM). A naive approach to analyzing the impact of known-key attacks would be to simply plug a certain block-cipher construction into a hash function and to analyze its security. However, as discussed in [MP15], this would be a devious and complex combinatorial task. The idea proposed in [MP15] is to "*model the block-ciphers in such a way that they behave randomly, except that an adversary can exploit a particular relation. More formally, we pose a certain predicate Φ, and we draw blockciphers randomly from the set of all ciphers that comply with predicate Φ. Throughout, we refer to this model as the 'weak cipher model (WCM)'.*"

In particular, consider the case in which the predicate Π implies for each key k the existence of a certain number A of sets of B queries $\{(k, p_1, c_1), ..., (k, p_B, c_B)\}$ - where $c_i = E_k(p_i)$ for each $i = 1, ..., B$ - that comply with a certain condition ϕ, defined as

$$\text{Bit}_C\big(p_1 \oplus ... \oplus p_B \oplus c_1 \oplus ... \oplus c_B\big) = 0 \tag{8.1}$$

where $Bits_C(x)$ outputs a string consisting of all bits of x whose index is in C. This model allows to cover several known-key distinguisher in the literature. We refer to [MP15] for a formal description of such a model.

***Generic* Known-Key Distinguishers.** In [MP15], authors limit themselves to consider a condition/property Φ defined as in (8.1). What about other possible conditions/properties that can be exploited by a know-key distinguisher?

Informally, a known-key distinguisher exploits the fact that it is in general harder for an adversary who does not know the key to derive an N-tuple of input blocks of the considered block cipher E that is "abnormally correlated" with the corresponding N-tuple of output blocks than for one who

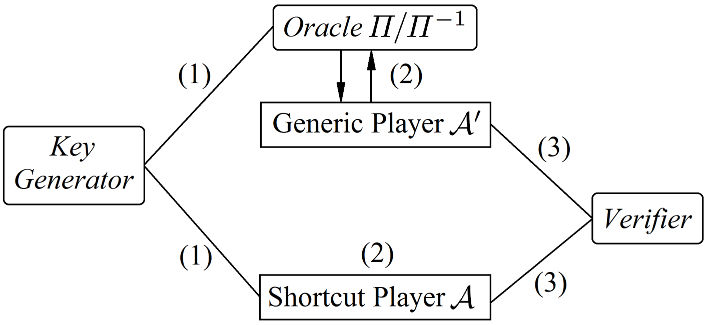


Figure 8.1.: A *Known-Key Distinguisher Scenario*. Step (0): a relationship $\mathcal{R}$ is chosen. Step (1): the secret key is given to the Oracle Π/Π^{-1}, to the Shortcut Player $\mathcal{A}$ and to the Verifier. Step (2): the Shortcut Player $\mathcal{A}$ and the Generic Player $\mathcal{A}'$ generate the N-tuples that satisfy the required relationship $\mathcal{R}$. Step (3): the Verifier receives the N-tuple and checks if $\mathcal{R}$ is satisfied or not. The faster player to generate the N-tuple wins the “game”.

knows the secret key. This difficulty is well expressed by the T-intractable definition, expressed by Gilbert [Gil14] as follows:

Definition 16 ([Gil14]). *Let $E : (K, X) \in \{0,1\}^k \times \{0,1\}^n \to E_K(X) \in \{0,1\}^n$ denote a block cipher of block size n bits. Let $N \geq 1$ and $\mathcal{R}$ denote an integer and any relation over the set S of N-tuples of n-bit blocks. $\mathcal{R}$ is said to be T-intractable relatively to E if, given any algorithm $\mathcal{A}'$ that is given an oracle access to a perfect random permutation Π of $\{0,1\}^n$ and its inverse, it is impossible for $\mathcal{A}'$ to construct in time $T' \leq T$ two N-tuples $\mathcal{X}' = (X'_i)$ and $\mathcal{Y}' = (Y'_i)$ such that $Y'_i = \Pi(X'_i)$, $i = 1, ..., N$ and $\mathcal{X}' \mathcal{R} \mathcal{Y}'$ with a success probability $p' \geq 1/2$ over Π and the random choices of $\mathcal{A}'$. The computing time T' of $\mathcal{A}'$ is measured as an equivalent number of computations of E, with the convention that the time needed for one oracle query to Π or Π^{-1} is equal to 1. Thus if q' denotes the number of queries of $\mathcal{A}'$ to Π or Π^{-1}, then $q' \leq T'$.*

Definition 17 ([Gil14]). *Let $E : (K, X) \in \{0,1\}^k \times \{0,1\}^n \to E_K(X) \in \{0,1\}^n$ denote a block cipher of block size n bits. A known-key distinguisher $(\mathcal{R}, \mathcal{A})$ of order $N \geq 1$ consists of (1) a relation $\mathcal{R}$ over the N-tuples of n-bit blocks (2) an algorithm $\mathcal{A}$ that on input a k-bit key K produces in time $T_{\mathcal{A}}$, i.e. in time equivalent with $T_{\mathcal{A}}$ computations of E, an N-tuple $\mathcal{X} = (X_i)$ $i = 1, ..., N$ of plaintext blocks and an N-tuple $\mathcal{Y} = (Y_i)$ $i = 1, ..., N$ of ciphertext blocks related by $Y_i = E_K(X_i)$ and by $\mathcal{X} \mathcal{R} \mathcal{Y}$. The two following conditions must be met:*

- *The relation $\mathcal{R}$ must be $T_{\mathcal{A}}$-intractable relatively to E;*
- *The validity of $\mathcal{R}$ must be efficiently checkable.*

To formalize the last requirement, one can incorporate the time for checking whether two N-tuples are related by $\mathcal{R}$ in the computing time $T_{\mathcal{A}}$ of algorithm $\mathcal{A}$.

We emphasize that while the algorithm $\mathcal{A}$ takes a random key K as input, *the relation $\mathcal{R}$ satisfied by the N-tuples of input and output blocks constructed by $\mathcal{A}$ or $\mathcal{A}'$ is the same for all values of K (in other words, it is independent of K) and must be efficiently checkable without knowing K.*

The “Inside-Out Approach” and Similarity with the Secret-Key Distinguisher. Before going on, we briefly highlight the similarity with the secret-key distinguisher. In this case, the goal is to find a set of (chosen) plaintexts (respectively ciphertexts) for which the corresponding ciphertexts (resp. plaintexts) - generated by the cipher for which the key is secret - satisfy a property $\mathcal{R}$ with a different probability than in the case in which the ciphertexts are generated by a random permutation Π.

In the case of a known-key distinguisher (similar for the chosen-key one), it is possible to do more, since the key that defines the cipher is not secret anymore. Instead of choosing plaintexts, the idea is to choose *middle texts* for which the corresponding plaintexts and the corresponding ciphertexts satisfy a certain property $\mathcal{R}$. This approach is usually denoted as the *"inside-out" approach.* The comparison with the random permutation case is then done by e.g. considering the cost to generate plaintexts and ciphertexts that satisfy the same property $\mathcal{R}$.

8.1.1. The Known-Key Distinguisher Scenario

To better understand these definitions, we propose and describe in more detail a generic scenario for a known-key distinguisher, which is depicted in Fig. 8.1. This scenario is composed of five characters, which are a key generator, an oracle, two players and a verifier. We assume that the oracle is instantiated by an *ideal cipher* $\Pi : (k, p) \in \{0,1\}^k \times \{0,1\}^n \to c = \Pi(k, p) \in \{0,1\}^n$ *chosen uniformly from all block ciphers of this form*[2]. Equivalently, Π is chosen uniformly at random among all ciphers with a k-bit key and a n-bit input/output. Moreover, we assume that the verifier knows both E and Π.

First of all - *step (0)*, we assume that a relation $\mathcal{R}$ defined as in Def. 16 is chosen. At *step (1)*, the key generator generates a key, which is given to the oracle and to one of the two player. In the following, we call "*shortcut player*" the player that knows the key and "*generic player*" the player that does not know it. Referring to the previous definitions by Gilbert, the generic player can be identified with the algorithm $\mathcal{A}'$, while the shortcut player can be identified with the algorithm $\mathcal{A}$. At *step (2)*, the two players generate the N-tuple of (plaintexts, ciphertexts) which satisfy the required relation $\mathcal{R}$. Since the generic player does not know the key, he must ask the oracle (identified with Π and/or Π^{-1} in the previous definitions) for the encryption (resp. decryption) of chosen plaintexts (resp. ciphertexts). We stress that this step does not consist only on the generation of (plaintext, ciphertext) pairs, but also includes any computational cost that the player must do in order to find the N-tuple with the required property. When a player finds the N-tuple which satisfies the required relation $\mathcal{R}$, he sends it to the verifier - *step (3)*. The verifier finally checks if (1) the relation $Y'_i = E_K(X'_i)$ (case of shortcut player) or $Y'_i = \Pi(X'_i)$ (case of generic player) is satisfied for each i and if (2) the N-tuple satisfied the relation $\mathcal{R}$. The first/fastest player who sends the N-tuple with the required property wins the "game".

A distinguisher is meaningful if the cost of the generic player - assume that the cost of one oracle-query is equal to the cost of one encryption - to generate the N-tuple is higher than the cost of the shortcut player, when the probability of success is equal for the two players. Equivalently, a distinguisher is meaningful if the probability of the generic player to win the game is higher than the probability of the shortcut player, when the number of tuples of (plaintexts, ciphertexts) that the two players can generate is fixed and equal for both players. In other words, in the first case one considers the computational costs of the two players to generate the N-tuples with a fixed probability of success (equal for both the players). In the second case, the computational cost (equivalent to the number of oracle queries for the generic player and the number of N-tuple generated by the shortcut one) is fixed and one considers the probabilities of success of the two players to win the game.

Role of the Verifier. *The role of the verifier is only to prevent one or both of the two players from cheating.* In other words, in the case of honest players, the verifier can be omitted, and the winner of the game is simply the first/fastest player that claims to have found the N-tuple of (plaintexts,

[2] In particular, for each fixed $k \in \{0,1\}^k$, $\Pi(k, \cdot)$ is a permutation, i.e. $\exists \Pi^{-1}(k, \cdot) : \{0,1\}^n \to \{0,1\}^n$ s.t. $\Pi^{-1}(k, \Pi(k, \cdot)) = \mathbb{I}(\cdot)$ where $\mathbb{I}(\cdot)$ is the identity function. The parameters k and n are the same that defines the encryption scheme E, that is $E : (K, p) \in \{0,1\}^k \times \{0,1\}^n \to c = E_K(p) \in \{0,1\}^n$.

ciphertexts) which satisfy the required relation $\mathcal{R}$. We highlight that such *a verifier is implicitly present in all the distinguishers currently present in the literature.*

In some distinguishers proposed in the literature, the computational cost of the verification step is not negligible. To clarify, we identify *the verification cost only as the cost to check the relation* $\mathcal{R}$[3]. Thus, we define *the cost of the distinguisher as the sum of* the cost of the verification step (i.e. *the cost of the verifier*) *and* of the cost to construct the set of plaintexts/ciphertexts with the required property (that is, *the cost of the shortcut player* - the cost of the other player is higher). For this reason, we assume in the following that *a relationship $\mathcal{R}$ is efficiently checkable if and only if the computational cost of the verifier is negligible with respect to the player ones.* This implies that the cost of the distinguisher can be approximated with the computational cost of the shortcut player (the cost of the other player is always higher). Moreover, this assumption prevents the construction of meaningless known-key distinguishers.

Generic Player. Since the generic player depends on the oracle to generate the N-tuple (i.e. he cannot work alone to generate it), two possible settings can be analyzed. In the first one, only the number of oracle queries is considered to determine the computational cost of this player, that is the number of encryptions/decryptions required by the generic player to the oracle. In the second one, both the number of oracle queries and any other computational cost of the generic player (which is in general not negligible) are considered. Intuitively this second setting is weaker than the first one, in the sense that a known-key distinguisher in the first setting works also in the second one but not vice-versa. In other words, one can expect that the required number N of tuples is higher in the first setting than in the second one (or equal in the best case). If the total cost of the generic player is well approximated by the number of queries, these two settings are completely equivalent.

8.1.2. Open Problem - How to Formally Define the "*Weak* Known-Key" Distinguisher?

From the above description, we can formulate a (potential) definition for the Known-Key disitinguisher in the "weak cipher model", denoted in the following as "Weak Known-Key distinguisher". We recall that the term "Weak" refer to the fact that we restrict the set of distinguishers only to those that exploit a property which is independent of the details of the underlying permutation.

In particular, *characterizing a meaningful - or non-trivial - known-key distinguisher for a concrete cipher E remains an open problem. Informally, a known-key distinguisher can be considered meaningful if the description of the generic relation $\mathcal{R}$ has no "obvious connection" with the specification of E and is independent of the value of the key. More generally, the relation $\mathcal{R}$ should not "extensively" re-use the operations that define E.* For the follow-up, we introduce a set $\mathfrak{D}$ of distinguishers D defined as following:

$\mathfrak{D}$ **Set of Distinguishers:** *$\mathfrak{D}$ denotes the set of all distinguishers D for which the description of the generic relation $\mathcal{R}$ has no "obvious connection" with the specification of E and it is independent of the value of the key.*

We emphasize that *the problem to formalize - with a proper mathematical definition - the set $\mathfrak{D}$ of all distinguishers D for which the description of the generic relation $\mathcal{R}$ has no "obvious connection" with the specification of E is still open for future research.* On the other hands, since our work is more "practically oriented" (i.e. we deal with E which is not an ideal cipher), in the following we limit ourselves to exploit this definition in order to set up known- (and chosen-) key distinguishers for concrete ciphers.

[3] In other words, the cost to check that the relation $Y_i' = E_K(X_i')$ (case of shortcut player) or $Y_i' = \Pi(X_i')$ (case of generic player) is satisfied for each i is *not* considered/included. In the following, we assume that such relations are always satisfied.

For a concrete example, note that a distinguisher that exploit the relation $X\mathcal{R}Y$ as $Y = E_X(X)$ does not belong to $\mathfrak{D}$. Instead, a distinguisher that exploits the relation $(X_1, X_2)\mathcal{R}(Y_1, Y_2)$ as $X_1 \oplus X_2 \in \mathcal{X}$ and $Y_1 \oplus Y_2 \in \mathcal{Y}$ for particular subspaces $\mathcal{X}$ and $\mathcal{Y}$ (equivalently, X_1 and X_2 are equal in certain bits/bytes/words - similar for Y_1 and Y_2) belongs in $\mathfrak{D}$, since such relation does not exploit any detail of $E(\cdot)$. In this last case, note that the texts $X = (X_1, X_2)$ satisfy a property which is independent of the property satisfied by the texts $Y = (Y_1, Y_2)$. In other words, given a set of texts X and Y, for almost all open-key distinguishers it happens that the relation $X\mathcal{R}Y$ is satisfied if and only if X satisfies a particular property Φ_X and Y satisfies a particular property Φ_Y. We emphasize that this approach is analogous to the one proposed e.g. in [MP15].

With this in mind, we can define the "weak known-key indifferentiability" that we are going to use in the following.

Definition 18. *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be block cipher (where $(K,p) \mapsto c = E(K,p) = E_K(p)$, and let Π an ideal block cipher. Let $D \in \mathfrak{D}$ be a distinguisher with oracle access to a permutation and its inverse, and returning a single bit. The "weak known-key indifferentiability" weakInf-KK advantage of D is defined as*

$$\begin{aligned} Adv^{weakInf\text{-}KK}(D) =& \big|Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{E_K(\cdot), E_K^{-1}(\cdot)}(K) = 1\big] + \\ & - Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{\Pi(K,\cdot), \Pi^{-1}(K,\cdot)}(K) = 1\big]\big|. \end{aligned}$$

For integers q_D and t, the weakInf-KK advantage of E is defined as

$$Adv^{weakInf\text{-}KK}(q_D, t) = \max_{D \in \mathfrak{D}} Adv^{weakInf\text{-}KK}(D)$$

where the maximum is taken over all distinguishers (for which the description of the generic relation $\mathcal{R}$ has no "obvious connection" with the specification of E) making at most q_D oracle queries and running in time at most t. E is a (q, t, ε) weakInf-KK if $Adv^{weakInf\text{-}KK}(q_D, t) \le \varepsilon$.

Note that, *even if the first probability does not contain any randomness, there's a time complexity involved in D.*

For the follow-up, we remember the definition of Pseudo-Random Permutation.

Definition 19. *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be block cipher (where $(K,p) \mapsto c = E(K,p) = E_K(p)$, and let Π an ideal block cipher. Let D be a distinguisher with oracle access to a permutation and its inverse, and returning a single bit. The (Strong PseudoRandom Permutation) SPRP-advantage of D is defined as*

$$Adv^{SPRP}(D) = \big|Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{E_K(\cdot), E_K^{-1}(\cdot)} = 1\big] - Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{\Pi(K,\cdot), \Pi^{-1}(K,\cdot)} = 1\big]\big|.$$

For integers q_D and t, the SPRP-advantage of E is defined as

$$Adv^{SPRP}(q_D, t) = \max_{D} Adv^{SPRP}(D)$$

where the maximum is taken over all distinguishers making at most q_D oracle queries and running in time at most t. E is a $(q, t, \varepsilon)-$SPRP if $Adv^{SPRP}(q_D, t) \le \varepsilon$.

Using these definitions, it turns out that *if a cipher is a Strong PseudoRandom Permutation, then the ideal cipher in the weakInf-KK definition can be replaced by the encryption scheme instantiated with an unknown secret key.* Informally, the ideal cipher is indistinguishable from the block cipher for which the key has been chosen at random.

Proposition 18. *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher which satisfies the SPRP ("Strong Pseudo-Random Permutation") definition. Then, E is (q,t,ε) weakInf-KK if and only if*

$$\max_{D\in\mathfrak{D}}\big|Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{E_K(\cdot),E_K^{-1}(\cdot)}(K)=1\big]-Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{E_K(\cdot),E_K^{-1}(\cdot)}=1\big]\big|\leq \varepsilon$$

where the maximum is taken over all distinguishers making at most q_D oracle queries and running in time at most t.

Proof. First, we prove that if E is (q,t,ε) weakInf-KK, then the claim holds:

$$\begin{aligned}
&\big|Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{E_K(\cdot),E_K^{-1}(\cdot)}(K)=1\big]-Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{E_K(\cdot),E_K^{-1}(\cdot)}=1\big]\big|\leq\\
\leq&\big|Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{E_K(\cdot),E_K^{-1}(\cdot)}(K)=1\big]-Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{\Pi(K,\cdot),\Pi^{-1}(K,\cdot)}(K)=1\big]\big|+\\
+&\big|Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{E_K(\cdot),E_K^{-1}(\cdot)}=1\big]-Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{\Pi(K,\cdot),\Pi^{-1}(K,\cdot)}=1\big]\big|
\end{aligned}$$

The second term in the l.h.s. is smaller than ε since E is a SPRP. It follows that

$$\begin{aligned}
&\big|Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{E_K(\cdot),E_K^{-1}(\cdot)}(K)=1\big]-Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{E_K(\cdot),E_K^{-1}(\cdot)}=1\big]\big|\leq\\
\leq&\big|Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{E_K(\cdot),E_K^{-1}(\cdot)}(K)=1\big]-Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{\Pi(K,\cdot),\Pi^{-1}(K,\cdot)}(K)=1\big]\big|+\varepsilon.
\end{aligned}$$

Using the same strategy, one can prove that

$$\begin{aligned}
&\big|Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{E_K(\cdot),E_K^{-1}(\cdot)}(K)=1\big]-Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{\Pi(K,\cdot),\Pi^{-1}(K,\cdot)}(K)=1\big]\big|\leq\\
\leq&\big|Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{E_K(\cdot),E_K^{-1}(\cdot)}(K)=1\big]-Prob\big[K \xleftarrow{\$} \{0,1\}^k; D^{E_K(\cdot),E_K^{-1}(\cdot)}=1\big]\big|+\varepsilon.
\end{aligned}$$

Since the previous two equalities work for all $\varepsilon \geq 0$, the thesis follows. □

Moreover, we show that if a cipher is weakInf-KK secure, then it is also SPRP secure.

Proposition 19. *If $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ is weakInf-KK secure, then it also satisfies the SPRP ("Strong Pseudo-Random Permutation") definition.*

Informally, *if it is not possible to distinguish $E_K(\cdot)$ from Π when the key K is known, then it is not possible to distinguish them when the key is secret.* Vice-versa is not true in general.

As example, the best secret-key distinguisher on AES (which is independent of the key) covers 6 rounds, while the best known-key distinguisher covers 8 rounds (12 if one allows Gilbert's strategy).

Proof. We are going to prove that if E is not SPRP, then it is not inf-KK secure.

If E is not SPRP, then by definition there exists a distinguisher $\hat{D}$ such that

$$\begin{aligned}
Adv^{SPRP}(q_D,t) \geq&Adv^{SPRP}(\hat{D}) = \big|Prob\big[K \xleftarrow{\$} \{0,1\}^k; \hat{D}^{E_K(\cdot),E_K^{-1}(\cdot)}=1\big]+\\
&- Prob\big[K \xleftarrow{\$} \{0,1\}^k; \hat{D}^{\Pi(K,\cdot),\Pi^{-1}(K,\cdot)}=1\big]\big|\geq \varepsilon.
\end{aligned}$$

The idea is to build an weakInf-KK distinguisher D using $\hat{D}$ that has the same advantage in breaking E. Distinguisher D simulates the environment for $\hat{D}$ as follows: firstly, a random key $K \xleftarrow{\$} \{0,1\}^k$ is selected uniformly and D runs on the input K; then it forward all queries by $\hat{D}$ - which is independent of K - to its own oracle. If $\hat{D}$ succeeds in distinguishing E and π, then D succeeds as well. In particular, we have $Adv^{weakInf\text{-}KK}(q_D,t) \geq Adv^{weakInf\text{-}KK}(D) = Adv^{SPRP}(\hat{D}) \geq \varepsilon$. □

Remark. Since in the following we only consider *practical* known-key distinguisher in the weak cipher model, we simply refer to them as "known-key distinguishers" instead of "weak known-key distinguishers".

Table 8.1.: *AES Known-Key Distinguishers.* The computation cost is the sum of the computational cost to generate N-tuples of plaintexts/ciphertexts and the verification cost. The word "Extended" refers to a distinguisher which exploits the technique introduced by Gilbert [Gil14] (in this case we also highlight which distinguisher is extended).

Rounds	Computations	Memory	Property	Reference
7	2^{56}	2^{56}	Zero-Sum	[KR07]
7	2^{24}	2^{16}	Differential Trail	[MRST09; LMR+09]
7	$\mathbf{2^{20}}$	$\mathbf{2^{16}}$	**Multiple Diff. Trail**	**[GR17]**
8	2^{64}	2^{64}	Uniform Distribution	[Gil14]
8	2^{48}	2^{32}	Differential Trail	[GP10]
8	2^{44}	2^{32}	Multiple Diff. Trail	[JNP13]
8	$2^{42.6}$	2^{13}	Statistical Integral	[CSCW17]
8	$\mathbf{2^{23}}$	$\mathbf{2^{16}}$	**Extended 7-Round MultDT**	**[GR17]**
9	$\mathbf{2^{50}}$	$\mathbf{2^{32}}$	**Extended 8-Round MultDT**	**[GR17]**
9	$\mathbf{2^{23}}$	$\mathbf{2^{16}}$	**Extended 7-Round MultDT**	**[GR17]**
10	2^{64}	2^{64}	Extended 8-Round Unif. Dist.	[Gil14]
10	$2^{59.6}$	2^{59}	Extended 8-Round Stat. Integral	[CSCW17]
10	$\mathbf{2^{50}}$	$\mathbf{2^{32}}$	**Extended 8-Round MultDT**	**[GR17]**
12	$\mathbf{2^{82}}$	$\mathbf{2^{32}}$	**Extended 8-Round MultDT**	**[GR17]**
12	$\mathbf{2^{66}}$	$\mathbf{2^{64}}$	**Extended 8-Round Unif. Dist.**	**[GR17]**

MultDT: Multiple Differential Trail

8.2. Known-Key Distinguishers for AES

In the following, we recall the known-key distinguishers present in the literature in the above scenario using the subspace trail notation. For simplicity, we assume that the relation $Y_i' = E_K(X_i')$ (case of shortcut player) or $Y_i' = \Pi(X_i')$ (case of generic player) are always satisfied for each i, that is that the two players do not cheat about this relations.

8.2.1. 7- and 8-Round Known-Key Distinguisher

In the 7- and 8-round known-key distinguishers proposed in [MRST09; LMR+09; LMS+15] and [GP10], the goal of the two players is to find two pairs of (plaintexts, ciphertexts) - i.e. (p^1, c^1) and (p^2, c^2) - with the following properties: the two plaintexts belong to the same coset of $\mathcal{D}_i$ - i.e. $p^1 \oplus p^2 \in \mathcal{D}_i$ - and the two ciphertexts belong to the same coset of $\mathcal{M}_i$ - i.e. $c^1 \oplus c^2 \in \mathcal{M}_i$ - for a fixed $i \in \{0, 1, 2, 3\}$.

In the above known-key distinguisher setting, the best technique that the shortcut player (i.e. the player who knows the key) can use to win the game is the *Rebound Attack.* The rebound attack is a differential attack proposed in [MRST09; LMR+09; LMS+15] for the cryptanalysis of AES-based hash functions. Since it is a differential attack, one needs a "good" (truncated) differential trail in order to exploit it. Examples of truncated differential trails used for 7- and 8-round AES are depicted in Fig. 8.2. The rebound attack consists of two phases, called inbound and outbound phase. In the first one, the attacker uses the knowledge of the key to find pairs of texts that satisfy the middle rounds of the truncated differential trail. In the second one, he propagates the solutions found in the first phase in the forward and in the backward directions, and checks if at least one of them satisfies the entire differential trail.

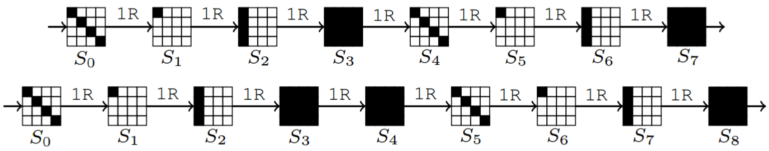


Figure 8.2.: 7- and 8-round differential characteristic for known-key distinguisher of AES-128.

As proved in [GP10], in the case of a perfect random permutation 2^{64} operations are required to find (plaintexts, ciphertexts) pairs (p_1, c_1) and (p_2, c_2) that have the required properties with good probability. Instead, for the AES case and using the rebound attack, 2^{48} computations are sufficient to find them with the same probability (besides a memory cost of $16 \times 2^{32} = 2^{36}$ bytes).

8.2.2. Multiple Limited-Birthday 8-Round Known-Key Distinguisher

An improvement of the previous known-key distinguisher on 8-round of AES was proposed in [JNP13]. Using the subspace trail notation, in this modified version of the 8-round known-key distinguisher, the goal of the two players is to find two pairs of (plaintexts, ciphertexts) such that the two plaintexts belong to the same coset of $\mathcal{D}_i$ for an arbitrary i and the two ciphertexts belong to the same coset of $\mathcal{M}_j$ for an arbitrary j, where i and j are not fixed in advance and it is not required that they are equal (i.e. no condition is imposed on i and j) - an example is given in Fig. 8.3. For arbitrary initial and final subspaces, the computational cost is reduced from 2^{48} to 2^{44} (note that there are 4 initial and final different subspaces $\mathcal{D}_i$ and $\mathcal{M}_j$, for a total of $4^2 = 2^4$ possibilities) while the required memory is still 2^{32}, as shown in detail in [JNP13].

8.3. Gilbert's Known-Key Distinguisher for 10-round AES

8.3.1. Uniform Distribution 8-round Known-Key Distinguisher

Another 8-round known-key distinguisher for AES is based on the uniform distribution property and it was proposed by Gilbert in [Gil14]. In this case, the goal of the two players is to find a set of 2^{64} (plaintext, ciphertext) pairs - that is (p^i, c^i) for $i = 0, ..., 2^{64} - 1$ - with the following properties:

- for each $K \subseteq \{0, 1, 2, 3\}$ with $|K| = 3$ the plaintexts are uniformly distributed in cosets of the diagonal space $\mathcal{D}_K$ - equivalently, for each K with $|K| = 3$ and for each $a \in \mathcal{D}_K^{\perp}$ there are 2^{32} plaintexts p^j for $j \in J \subseteq \{0, ..., 2^{64} - 1\}$ with $|J| = 2^{32}$ such that $p^j \in \mathcal{D}_K \oplus a$ for all $j \in J$;

- for each $K \subseteq \{0, 1, 2, 3\}$ with $|K| = 3$ the ciphertexts are uniformly distributed in cosets of the mixed space $\mathcal{M}_K$ - equivalently, for each K with $|K| = 3$ and for each $a \in \mathcal{M}_K^{\perp}$ there are 2^{32} ciphertexts c^j for $j \in J \subseteq \{0, ..., 2^{64} - 1\}$ with $|J| = 2^{32}$ such that $c^j \in \mathcal{M}_K \oplus a$ for all $j \in J$.

If the final MixColumns is omitted, an equivalent condition holds on the ciphertexts by replaying the mixed space $\mathcal{M}_K$ with the inverse-diagonal one $\mathcal{ID}_K$. To be more formal:

Definition 20. *Consider 2^{64} texts $t^i \in \mathbb{F}_{2^8}^{4\times4}$ for $i = 0, ..., 2^{64} - 1$, and let $K \subseteq \{0, 1, 2, 3\}$ with $|K| = 3$ fixed. We say that these 2^{64} texts t^i are "uniformly distributed" in cosets of $\mathcal{M}_K$ if*

- *for each coset $\mathcal{M}_K \oplus a$ for $a \in \mathcal{M}_K^{\perp}$, there exist 2^{32} texts $T_a = \{t^j\}_{j=0,...,2^{32}-1}$ such that $t^j \in \mathcal{M}_K \oplus a$ for each $t^j \in T$;*

- *given sets T_a and T_b just defined for two different cosets $\mathcal{M}_K \oplus a$ and $\mathcal{M}_K \oplus b$ where $(a \oplus b) \in \mathcal{M}_K^{\perp}$, then $T_a \cap T_b = \emptyset$.*

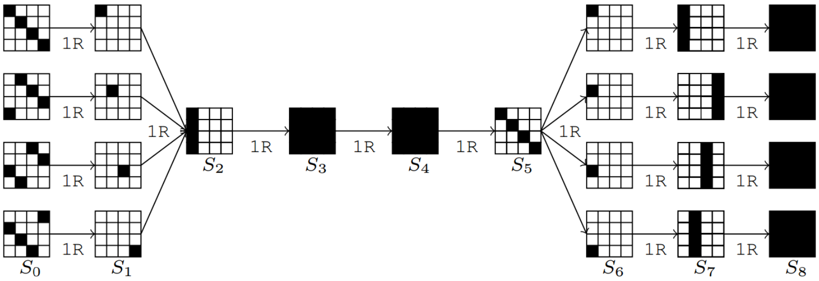


Figure 8.3.: 8-round multiple differential characteristics for known-key distinguisher of AES-128.

In the case in which the final MixColumns operation is omitted, note that it is possible to re-formulate the goal of the two players as following: find a set of 2^{64} (plaintext, ciphertext) pairs - that is (p^i, c^i) for $i = 0, ..., 2^{64} - 1$ - such that the bytes of the plaintexts and the ciphertexts are uniformly distributed, that is:

- for each $j, k = 0, 1, 2, 3$ and for each $x \in \mathbb{F}_{2^8}$, there are 2^{56} plaintexts p^i for $i \in I \subseteq \{0, ..., 2^{64}-1\}$ with $|I| = 2^{56}$ that satisfy $p^i_{j,k} = x$ for all $i \in I$;

- for each $j, k = 0, 1, 2, 3$ and for each $x \in \mathbb{F}_{2^8}$, there are 2^{56} ciphertexts c^i for $i \in I \subseteq \{0, ..., 2^{64} - 1\}$ with $|I| = 2^{56}$ that satisfy $c^i_{j,k} = x$ for all $i \in I$.

We prove that these two properties are equivalent for the ciphertexts (the same argumentation applies on the plaintexts as well).

First of all, if the bytes of the ciphertexts are uniformly distributed, then the ciphertexts are uniformly distributed in cosets of the inverse-diagonal space $\mathcal{ID}_K$ for each K with $|K| = 3$ by definition of $\mathcal{ID}_K$. Vice-versa, consider the case in which the ciphertexts are uniformly distributed in cosets of $\mathcal{ID}_K$ for each K with $|K| = 3$. By definition, there are 2^{32} ciphertexts $\hat{c}^i$ with $i \in I \subseteq \{0, ..., 2^{64} - 1\}$ and $|I| = 2^{32}$ that belong to the same coset of $\mathcal{ID}_{0,1,2} \oplus a$ for a certain $a \in \mathcal{ID}^{\perp}_{0,1,2}$ (equivalent for the other spaces $\mathcal{ID}_K$ with $|K| = 3$). By definition, $a \in \mathcal{ID}^{\perp}_{0,1,2}$ if and only if $a_{k,j} = 0$ for $(k, j) \neq (0, 3), (1, 2), (2, 1), (3, 0)$, i.e. for each $k + j \neq 3$. In other words, $\hat{c}^i \in \mathcal{ID}_{0,1,2} \oplus a$ for each $i \in I$ if and only if $\hat{c}^i_{k,j} = a_{k,j}$ for each $i \in I$ and for each $k + j = 3$. Working independently on each byte, it follows that the bytes of c^i are uniformly distributed (for example, working on the first byte and considering all $a \in \mathcal{ID}^{\perp}_{0,1,2}$ with $a_{0,3}$ fixed, it follows that there are $2^{24} \cdot 2^{32} = 2^{56}$ ciphertexts c^i s.t. $c^i_{0,3} = a_{0,3}$). If the final MixColumns is not omitted, the goal of the two players becomes to find a set of 2^{64} (plaintext, ciphertext) pairs - that is (p^i, c^i) for $i = 0, ..., 2^{64} - 1$ - such that the bytes of the p^i and of $MC^{-1}(c^i)$ are uniformly distributed.

Finally we highlight that *the uniform distribution implies the balance/zero-sum property* both on the plaintexts and on the ciphertexts, and that the balance property is not destroyed by the (final) MixColumns operation (since this operation is linear). For completeness, we remember that texts $\{t^i\}_{i \in I}$ have the balance property if $\bigoplus_{i \in I} t^i = 0$.

The Strategy of the Shortcut Player. Here, we briefly recall the best strategy that the shortcut player can use to win the game using the subspace trails notation[4]. The idea is to start in the middle with a set S of texts defined as $S := \mathcal{D}_i \oplus \mathcal{M}_j \oplus c$ for a constant c, where $|S| = 2^{64}$. Observe that

$$\mathcal{D}_i \oplus \mathcal{M}_j \oplus c \equiv \bigcup_{b \in \mathcal{D}_i \oplus c} \mathcal{M}_j \oplus b = \bigcup_{a \in \mathcal{M}_j \oplus c} \mathcal{D}_i \oplus a,$$

[4] We mention that the same strategy is described using the *super-Sbox* notation in [Gil14].

i.e. the set S can be re-written as union of cosets of the space $\mathcal{D}_i$ or as union of cosets of the space $\mathcal{M}_j$. The ciphertexts are given by the 4-round encryption of S, while the plaintexts by the 4-round decryption of S.

After encrypting S for 4 rounds, the texts are uniformly distributed in each coset of $\mathcal{M}_I$ of dimension 12 (i.e. $|I| = 3$). That is, after 4 rounds, each coset of $\mathcal{M}_I$ for $|I| = 3$ contains exactly 2^{32} elements. Indeed, remember that given two elements in the same coset of $\mathcal{D}_I$, they can not belong to the same coset of $\mathcal{M}_J$ for $|I| + |J| \leq 4$ after 4-round. Thus, given a coset of $\mathcal{D}_i$ with $|i| = 1$, after 4 rounds each element is distributed in a different cosets of $\mathcal{M}_J$ for $|J| = 3$. Since a coset of $\mathcal{D}_i$ contains 2^{32} elements and since there are exactly 2^{32} cosets of $\mathcal{M}_J$, the elements of $\mathcal{D}_i \oplus \mathcal{M}_j$ are uniformly distributed in each coset of $\mathcal{M}_I$. The same happens if one decrypts S for 4 rounds. In this case, after decrypting S for 4 rounds, the texts are uniformly distributed in each coset of $\mathcal{D}_I$ of dimension 12 (i.e. $|I| = 3$), that is each coset of $\mathcal{D}_I$ for $|I| = 3$ contains exactly 2^{32} elements.

On the Meaningfulness of this Distinguisher. For the follow-up, we briefly recall the argumentation given by Gilbert about the meaningfulness of such distinguisher.

First of all, 2^{64} texts satisfy the uniform distribution on each byte with probability

$$p = \left[\prod_{i=0}^{255} \binom{2^{64} - i \cdot 2^{56}}{2^{56}} \cdot \left(2^{-8}\right)^{2^{64}}\right]^{16}.$$

Indeed, consider the following problem. Given N texts and 2 sets, assume that each text belongs to one of the two sets with probability 2^{-1}. It follows that the N texts are uniformly distributed among the two sets with prob. $\binom{N}{N/2} \cdot 2^{-N}$. In a similar way, given $d \geq 2$ sets, they are uniformly distributed with probability[5] $\left(\prod_{i=0}^{d-1} \binom{N - i \cdot N/d}{N/d} \cdot d^{-N}\right)$.
Using Stirling's formula $n! \simeq n^n \cdot e^{-n} \cdot \sqrt{2\pi \cdot n}$, this probability is well approximated by

$$p = \left(\frac{2^{64}!}{(2^{56}!)^{256}} \cdot \left(2^{-8}\right)^{2^{64}}\right)^{16} \simeq \left(\frac{1}{2^{49} \cdot \pi}\right)^{128} \cdot (256!)^{-1/2} \simeq 2^{-7328.1} \equiv 2^{-2^{12.84}}. \qquad (8.2)$$

In other words, given 2^{64} plaintexts whose bytes are uniformly distributed, this represents the probability that the bytes of the corresponding ciphertexts are uniformly distributed. For comparison, given 2^{64} plaintexts whose sum is zero, then the sum of the corresponding ciphertexts is equal to zero with probability 2^{-128}.

What is the minimum number $N \equiv 2^{64} + M > 2^{64}$ of - random - (plaintext, ciphertext) pairs such that there is a subset of 2^{64} pairs whose bytes are uniformly distributed both on the plaintexts and on the ciphertexts with non-negligible property? Given $2^{64} + M$ texts, it is possible to construct

$$\binom{2^{64} + M}{2^{64}} \simeq \frac{1}{\sqrt{2\pi \cdot M}} \cdot \left(\frac{2^{64} + M}{M}\right)^{M}$$

different sets of 2^{64} texts (where the approximation is given using the Stirling's formula and by the assumption $M \ll 2^{64}$). This number is always higher than $p^{-2} \equiv 2^{2^{13.84}}$ for each $M \geq 2^{12}$. In other words, given $2^{64} + 2^{12}$ random pairs, there is a good probability to find 2^{64} (plaintext, ciphertext) pairs such that the bytes of the plaintexts and of the ciphertexts are uniformly distributed. It follows that if the cost of the generic player is approximated by the number of oracle queries, then his cost is approximately of $2^{64} + 2^{12} \simeq 2^{64}$ encryptions vs 2^{64} encryption of the shortcut player.

[5] Consider the case $N = 2^{64}$ and $d = 256$. The product of the binomial coefficients is explained as follows. For each one of the 16 bytes, there must exist $2^{64}/256 = 2^{56}$ texts for each one of the 256 possible values. Thus, there are $\binom{2^{64}}{2^{56}}$ possible sets of 2^{56} texts for each the byte as value 0, $\binom{2^{64} - 2^{56}}{2^{56}}$ possible sets of 2^{56} texts for each the byte as value 1 and so on.

So, *why is this distinguisher meaningful?* Instead of focusing on the cost of the players, the idea is to consider the probability of the generic player to win the game given 2^{64} texts is negligible. To do this, authors of [Gil14] claim that this probability is upper bounded by the probability of the following game: *"given* $2^{64}-1$ *(plaintext, ciphertext) pairs whose bytes are 'almost uniform'* - see the definition in the following, *find a text for which the bytes of the corresponding* 2^{64} *texts are uniformly distributed"*. Since this probability is upper bounded by 2^{-127} - see proof of Prop. 4 of [Gil14] - and since this second game is (strong) "related" to the original one[6], the conclusion follows immediately.

For completeness, we formal define what "almost uniform" means. Consider $2^{64}-1$ texts $t^i \in \mathbb{F}_{2^8}^{4\times 4}$ for $i = 0, ..., N-2$. We say that the bytes of $2^{64}-1$ texts t^i are "*almost uniform*" if for each row and column $j, k = 0, 1, 2, 3$ (1) there exists $x \in \mathbb{F}_{2^8}$ s.t. there are $2^{56}-1$ texts that satisfy $t^i_{j,k} = x$ and (2) for each $y \in \mathbb{F}_{2^8} \setminus x$, there are 2^{56} texts that satisfy $t^i_{j,k} = y$. More generally:

Definition 21. *Consider* $2^N - d$ *texts* $t^i \in \mathbb{F}_{2^8}^{4\times 4}$ *for* $i = 0, ..., N-d-1$ *for* $d \geq 1$*. The bytes of these* $2^N - d$ *texts* t^i *are "almost uniform" if for each row and column* $j, k = 0, 1, 2, 3$*:*

- *there exists a set* $X \equiv \{x_1, ..., x_s \in \mathbb{F}_{2^8}\}$ *with cardinality* $s \leq d$ *such that for each* $x_l \in X$ *with* $1 \leq l \leq s$ *there are* $2^{N-8} - d \leq \hat{s}_l \leq 2^{N-8} - s$ *texts that satisfy* $t^i_{j,k} = x_l$ *where* $\sum_{l=1}^{s} \hat{s}_l = d$*;*
- *for each* $y \in \mathbb{F}_{2^8} \setminus X$*, there are* 2^{N-8} *texts that satisfy* $t^i_{j,k} = y$*.*

Proposition 20. *Consider a set of* 2^N *texts whose bytes are uniformly distributed. For each* $d \geq 1$*, the bytes of each subset of* $2^N - d$ *texts are "almost uniform" distributed w.r.t. the previous definition.*

8.3.2. Extension to 10 Rounds of AES

This distinguisher is the starting point used by Gilbert in order to set up the first 10-round known-key distinguisher for AES. The basic idea is to extend this 8-round distinguisher based on the uniform distribution property adding one round at the end and one at the beginning. Assume for simplicity that the final MixColumns is omitted. In the known-key distinguisher scenario presented above, the players have to send to the verifier 2^{64} (plaintext, ciphertext) pairs, that is (p^i, c^i) for $i = 0, ..., 2^{64}-1$, with the following properties[7]:

1. there exists a key k^0 s.t. the bytes of $\{R_{k^0}(p^i)\}_i$ are uniformly distributed, or equivalently that the texts $\{R_{k^0}(p^i)\}_i$ are uniformly distributed among the cosets of $\mathcal{D}_I$ for each I with $|I| = 3$;
2. there exists a key k^{10} s.t. the bytes of $\{MC^{-1} \circ R^{-1}_{k^{10}}(c^i)\}_i$ are uniformly distributed, or equivalently that the texts $\{R^{-1}_{k^{10}}(c^i)\}_i$ are uniformly distributed among the cosets of $\mathcal{M}_J$ for each J with $|J| = 3$;

where MC^{-1} denotes the inverse MixColumns operation. We emphasize that *it is not required that* k^0 *and* k^{10} *are equal to the secret subkeys, that is* k^r *can be different from the* r*-th subkey. In other words, it is only required that such keys exist, and not that they are equal to the "real" keys that defines* $E_K(\cdot)$*. The same assumption is exploited in all for all Gilbert's like distinguishers presented in the literature.* Moreover, in this game, the subkeys k^0 and k^{10} are assumed to be independent (argumentations are given by Gilbert to show that the same distinguisher is applicable also to the case in which the key-schedule holds - we discuss this topic in details in the following).

For completeness, note that *since uniform distribution implies balance property* – vice-versa is not true in general, if the previous properties are satisfied then – for the key k^0 – the sum of the

[6] For completeness, we mention that *no formal proof is provided* in [Gil14] in order to support this claim. In other words, it is not proved that the fact that this second game is "hard" implies the hardness of the original game, and/or vice-versa.

[7] For this and the following distinguishers, we abuse the notation k^r to denote a key of the a certain round r. In general, such subkey k^r is different from the real secret subkey.

plaintexts after one round is equal to zero, i.e. $\bigoplus_{i=0}^{2^{64}-1} R_{k^0}(p^i) = 0$, and – for the key k^{10} – the sum of the ciphertexts one round before is equal to zero, i.e. $\bigoplus_{i=0}^{2^{64}-1} R^{-1}_{k^{10}}(c^i) = 0$.

We emphasize that *even if this is a known-key distinguisher, the keys k^0 and k^{10} for which the relation $\mathcal{R}$ is satisfied can be different from the real subkeys. In other words, the verifier has no information of the keys for which the relation $\mathcal{R}$ is satisfied, and her task is to check if they exist.* It follows that one must show that the above conditions are efficiently checkable. The only way to verify these requirements is to find these two subkeys in an efficient way, which is not possible using a brute force attack (k^0 and k^{10} have 128 bits). Instead of checking all the $2 \cdot 2^{128} = 2^{129}$ possible values of k^0 and k^{10}, the idea proposed in [Gil14] is to check uniform distribution working on single columns of $SR(c^i)$ and of $SR^{-1}(p^i)$. In this way, the verifier must guess only 32 bits instead of 128, and she has to repeat this operation 4 times (one for each anti-diagonal/diagonal) for each key.

For completeness, we mention that another strategy can be used to check the required property. Working independently on each byte of k^0 and k^{10} instead of entire anti-diagonal/diagonal, the idea is simply to use integral attack [DKR97; KW02] to filter wrong keys. Besides improving the computational cost, this strategy allows also to extend the distinguisher on 12-round AES.

In conclusion, the shortcut player (i.e. the one who knows the key) can construct these 2^{64} (plaintext, ciphertext) pairs using the same strategy proposed for the 8 rounds distinguisher (note that in this case the keys k^0 and k^{10} correspond to the secret sub-keys). Instead, as proved in Prop. 6 of [Gil14], the probability that the generic player (i.e. the one who does not know the secret key) successfully outputs (input, output) pairs that satisfy the previous properties (both in the input and in the output) is upper bounded by $2^{-16.5}$. Finally, the verifier can find the keys k^0 and k^{10} that satisfy the required property (if they exist) with a computational cost which is smaller than the cost of the two players.

On the Meaningfulness of this Distinguisher. For the follow-up, we briefly recall the argumentation given in [Gil14] about the meaningfulness of this distinguisher.

First of all, what is the probability that given a set of 2^{64} texts there exists a key $\hat{k}$ such that the bytes of 1-round encryption (resp. decryption) of such texts are uniformly distributed? Using the previous calculation and since there are 2^{128} different keys, this probability is equal to $2^{128} \cdot p \simeq 2^{128} \cdot 2^{-7328.1} = 2^{-7200.1} \equiv 2^{-2^{12.81}}$ where p is defined in (8.2). Similar to the 8-round case, it follows that $2^{64} + 2^{12} \simeq 2^{64}$ (plaintext, ciphertext) pairs are sufficient to have good probability to win the game.

So, as before, *why is this distinguisher meaningful?* As for the 8-round case, instead of focusing on the cost of the players, Gilbert shows that the probability of the generic player to win the game given 2^{64} texts is negligible. To do this, Gilbert claims that this probability is upper bounded by the probability of the following game. Consider $2^{64} - d$ (plaintext, ciphertext) pairs for $d \geq 5$, that is (p^i, c^i) for each $i = 0, ..., 2^{64} - d - 1$, with the property that there exist a set of keys k^0 and k^{10} - this set can correspond to the entire set of keys - for which the bytes of $R_{k^0}(p^i)$ and of $MC^{-1} \circ R^{-1}_{k^{10}}(c^i)$ (that is 1-round encryption of p^i and the 1-round decryption of the ciphertexts) are "almost uniform" distributed. The goal of the player is to find the remaining d texts for which the bytes of the corresponding 2^{64} texts after 1-round encryption/decryption are uniformly distributed. Since this probability is upper bounded by $(2^{128})^2 \cdot \left(\frac{5^{16}}{2^{128}-2^{64}+1}\right)^3 \simeq 2^{-16.5}$ - see proof of Prop. 6 of [Gil14] - and since this second game is "related" to the original one, the conclusion follows immediately.

Generic Considerations

The previous 10-round distinguisher proposed by Gilbert is different from all the previous distinguishers up to 8 rounds present in the literature. For all distinguishers up to 8-round, the relation $\mathcal{R}$ that the N-tuple of (plaintexts, ciphertexts) must satisfy does not involve any operation of the block cipher E. As a consequence, it allows the verifier to check whether the N-tuple of (plaintexts, ciphertexts) satisfy the required relation $\mathcal{R}$ without knowing anything of the key. When $\mathcal{R}$ does not re-use operations of E, this provides some heuristic evidence that this distinguisher can be considered *meaningful*.

On the other hand, the previous 10-round distinguisher and the ones that we are going to propose do not satisfy this requirement, i.e. in these cases the relation $\mathcal{R}$ involves and re-uses some operations of E. The novelty of Gilbert's work is not just the possibility to extend the distinguisher up to 10-round AES, but rather the introduction of a new distinguisher model. Requiring the existence of round keys for which the 1-round encryption of the plaintexts (respectively, 1-round decryption of the ciphertexts) satisfy the relation $\mathcal{R}$, or in other words considering relations $\mathcal{R}$ that depend on some operations of E, allows to set up new distinguishers that penetrate more round of the block cipher.

8.3.3. Statistical Integral Distinguisher with Multiple Structures

Finally, we mention for completeness that at ACISP 2017 the distinguishers proposed by Gilbert in [Gil14] has been improved by T. Cui, L. Sun, H. Chen and M. Wang [CSCW17]. In this paper, authors turn both the 8- and 10-round Gilbert's distinguishers into "statistical integral ones" [WCC+16] with the goal to reduce the data/time complexity.

As we have previously recalled, the 8-round Gilbert's distinguisher is based on the uniformly distributed integral property, that is the goal is to generate plaintexts and corresponding ciphertexts that are uniform distributed respectively in cosets of $\mathcal{D}_J$ and $\mathcal{M}_I$ for each $I, J \subseteq \{0, 1, 2, 3\}$ with $|I| = |J| = 3$. This property is turned into a "statistical integral property" on each byte of input and output using the strategy proposed in [WCC+16]. Although the uniformly distributed property does not strictly hold in the statistical integral distinguisher, it is proved in [WCC+16] (see Prop. 1) that the distribution of input/output values for a cipher can be distinguished from the distribution of output values which originate from a random permutation. This 8-round distinguisher is then turned into a 10-round one using the same strategy proposed by Gilbert in [Gil14], that is requiring that an initial and a final keys exist such that the statistical integral property is satisfied by the plaintexts after one round of encryption and by the ciphertexts after one round of decryption.

8.4. Revisiting Gilbert's Distinguisher: is it a "Valid" Model?

In the conclusion of his paper, Gilbert claims that it seems technically difficult to use a stronger property than the uniform distribution one to extend an 8-round known-key distinguisher to a 10-round one:

1st Conjecture: "*while we do not preclude that the use of the stronger property that several pairs satisfying the differential relation of [GP10]* [i.e. truncated diff. relations exploited by the rebound distinguisher] *can be derived might potentially result in a 10-round distinguisher that outperforms the 10-round distinguisher presented above, giving a rigorous proof seems technically difficult.*"

In particular, he left "*the investigation of improved 10-round known-key distinguishers and associated proofs - or even plausible heuristic arguments if rigorous proofs turn out to be too difficult to obtain - as an open issue.*"

In our paper [GR17], we picked up this challenge, and using a strategy similar to the one proposed by Gilbert in [Gil14], we show how to construct more efficient 8-, 9- and 10-round distinguishers exploiting known-key distinguishers based on truncated differential trails. In particular, we use as starting point the 8-round known-key distinguisher presented in [JNP13], and we extend it at the end or/and at the beginning using the same strategy proposed by Gilbert. This allows to set

Table 8.2.: *1st/2nd Conjectures and AES Gilbert's Known-Key Distinguishers.* Referring to the 1*st* and the 2*nd* conjectures given in the main text, in this table we emphasize which ones of our - and others in teh literature - results disprove them. This is the starting point exploited in order to discuss the validity of Gilbert's model.

Rounds	Property	*1st Conjecture*	*2nd Conjecture*	Reference
9	***extended* 8-Round MultDT**	✓		**[GR17]**
9	***extended* 7-Round MultDT**	✓		**[GR17]**
10	*extended* 8-Round Unif. Dist.			[Gil14]
10	*extended* 8-Round Stat. Integral	✓		[CSCW17]
10	***extended* 8-Round MultDT**	✓		**[GR17]**
12	***extended* 8-Round MultDT**	✓	✓	**[GR17]**
12	***extended* 8-Round Unif. Dist.**		✓	**[GR17]**

MultDT: Multiple Differential Trail

up a 9-round known-key distinguisher and a 10-round known-key distinguisher for AES with time complexity approximately of 2^{50}.

As a main cryptanalytic results, we show that it is possible to extend our 10-round distinguisher up to 12 rounds and that it is possible to extend Gilbert's 10-round distinguisher based on the uniform distribution property up to 12 rounds. These are the *first known-key distinguisher for full AES-192*, and they also provide counter-examples of the claim made in [Gil14] about the (im)possibility to use Gilbert's technique to extend a 8-round distinguisher more than 2 rounds:

2nd Conjecture: "*The reader might wonder whether the technique we used to derive a known-key distinguisher for the 10-round AES from a known-key distinguisher for the 8-round AES does not allow to extend this 8-round known distinguisher by an arbitrary number of rounds. It is easy however to see that the argument showing that 10-round relation $\mathcal{R}$ is efficiently checkable does not transpose for showing that the relations over $r > 10$ rounds one could derive from the 8-round relation by expressing that the r-round inputs and outputs are related by $r - 8 > 2$ outer rounds to intermediate blocks that satisfy the 8-round relation are efficiently checkable.*"

In the following, we briefly recall our results presented in details in [GR17].

8.4.1. 10-round Distinguisher based on the Truncated Differential Trails

Using the same strategy proposed by Gilbert in [Gil14], we set up our 10-round distinguisher by extending the 8-round one presented in [JNP13] and in Sect. 8.2.2 both at the beginning and at the end.

In the above defined known-key distinguisher scenario, the players have to send to the verifier $n \geq 64$ different tuples of (plaintext, ciphertext) pairs, that is $\{(p_i^1, c_i^1), (p_i^2, c_i^2)\}$ for $i = 0, ..., n-1$, with the following properties:

1. there exists a key k^0 s.t. for each tuple there exists j for which the two plaintexts belong to the same coset of $\mathcal{D}_j$ after one round, that is
$$\exists\, k^0 \quad \text{s.t.} \quad \forall i = 0, ..., n-1, \quad \exists j \in \{0, ..., 3\} \quad \text{s.t.} \quad R_{k^0}(p_i^1) \oplus R_{k^0}(p_i^2) \in \mathcal{D}_j;$$
2. there exists a key k^{10} s.t. for each tuple there exists l for which the two ciphertexts belong to the same coset of $\mathcal{M}_l$ one round before, that is
$$\exists\, k^{10} \quad \text{s.t.} \quad \forall i = 0, ..., n-1, \quad \exists l \in \{0, ..., 3\} \quad \text{s.t.} \quad R^{-1}_{k^{10}}(c_i^1) \oplus R^{-1}_{k^{10}}(c_i^2) \in \mathcal{M}_l.$$

We stress that the keys k^0 and k^{10} must be equal for all the tuples, otherwise it is straightforwards to generate tuples with the required properties. In other words, if there exist two different tuples (c_0, c_1) and (c_2, c_3) such that $R_k^{-1}(c_0) \oplus R_k^{-1}(c_1) \in \mathcal{M}_l$ and $R_{\tilde{k}}^{-1}(c_2) \oplus R_{\tilde{k}}^{-1}(c_3) \in \mathcal{M}_{\tilde{l}}$ for two different keys $k \neq \tilde{k}$, then the above defined relationship $\mathcal{R}$ is not satisfied[8]. In particular, the claim "*the transposition of our technique to the 8-round distinguisher of [GP10] does not allow to derive a valid 10-round distinguisher*" made in [Gil14] to support the impossibility to set up such 10-round distinguisher based on truncated differential trails is justified only when no assumption on the key k is done. In other words, the above defined relationship $\mathcal{R}$ together with the requirement of *uniqueness of the key k* allows to extend the 8-round distinguisher of [GP10] as in [Gil14].

Before going on, it is also important to emphasize that no condition on the keys k^0 and k^{10} is imposed, except that they exist and they are equal for all the tuples. That is, *it is not required that this key is equal to the real secret subkey.* The same consideration holds also for the next distinguishers presented here, and for the 10-round distinguisher presented by Gilbert in [Gil14].

Moreover, since the verifier has to check the existence of both k^0 and k^{10}, two possible scenarios can be considered and studied:

1. no key-schedule holds - k^0 and k^{10} are independent;
2. AES key-schedule among k^0 and k^{10}.

Intuitively, the second case (i.e. with key schedule) is harder than the first one (i.e. without key schedule) for the generic player, since a further property must be verified. In other words, the time required by this player to generate the tuples for the second scenario is not lower than for the first one, that is the probability of success in the second scenario is not higher than in the first one.

In the following we briefly analyze why the distinguisher works – more details can be found in [GR17].

Shortcut Player. First of all, for the shortcut player, the two scenarios (with/without key schedule) are completely identical. Indeed, using the rebound technique, he is able to generate n tuples that satisfy all the conditions (included the key schedule without any additional cost). The computational cost of this player is well approximated by $n \cdot 2^{44}$ computations. In the case of independent subkeys, the computational cost of the generic player to generate n tuples that satisfy all the conditions is approximately of $n \cdot 2^{61.56-128/n}$ queries. As a result, if $n \geq 64$, then the cost of the generic shortcut is of 2^{50} computations.

Verifier. Using the truncated attack proposed in [GRR16], the cost of the verifier is well approximated by $2^{12.8}$ encryptions, much less than the cost of the two players.

Generic Player

Here we limit to consider the case of independent subkeys. The idea is to choose plaintexts such that the condition on the plaintexts is fulfilled with probability 1. To do this, the generic player must fix a random key $\hat{k}$, and computes for a certain $j \in \{0, ..., 3\}$ and for a random $a \in \mathcal{D}_j^{\perp}$ the following set:

$$D_a := R_{\hat{k}}^{-1}(\mathcal{D}_j \oplus a). \tag{8.3}$$

The idea is choose/use plaintexts in this set D_a just defined. In other words, the player works in the same way described for the 9-round distinguisher but using D_a defined above instead of a coset of

[8] Note that without this request e.g. on the secret key k^{10}, it is extremely easy to construct tuples such that the two ciphertexts belong to the same coset of $\mathcal{M}_l$ one round before. Indeed, given two ciphertexts c^1 and c^2, on average there exist $4 \cdot (2^8)^4 = 2^{34}$ different keys such that $R^{-1}(c^1) \oplus R^{-1}(c^2) \in \mathcal{M}_l$ for a certain l. Thus, it is straightforward to construct n different tuples with the above defined relationship $\mathcal{R}$ but without any condition on the key k^{10}. Similar considerations hold for the key k^0.

$\mathcal{D}_j$. The corresponding ciphertexts are simply got by oracle-queries. Since the cardinality of a coset of $\mathcal{D}_j$ is 2^{32}, the computation of a set D_a requires $2^{32+4} = 2^{36}$ S-Box look-ups for each coset $\mathcal{D}_j \oplus a$. Note that if the player needs more than 2^{32} (plaintext, ciphertext) pairs, he simply chooses another $a' \in \mathcal{D}_j^{\perp}$ (or/and another j) and, using the *same* key $\hat{k}$, he computes the corresponding set $D_{a'}$ defined as before. We emphasize that the player must use always the same key $\hat{k}$ to compute these sets, in order to fulfill the property on the plaintexts. We stress that given plaintexts in the same set D_a, the requirement on the plaintexts is always fulfilled since by construction there exists a key (which is $\hat{k}$) such that the plaintexts of each tuple belong to the same coset of $\mathcal{D}_j$ after one round.

Given the set D_a, the player asks the oracle for the corresponding ciphertexts. The idea is to check if there exists a key k and n tuples such that the two ciphertexts of each of these n tuples belong to the same coset of $\mathcal{M}_l$ one round before. We remember that it is not necessary that the key for which this condition is satisfied is the real one.

As shown in details in [GR17], given a single tuple there exist on average 2^{34} keys such that the two ciphertexts belong to the same coset of $\mathcal{M}_j$ one round before. To set up a meaningful distinguisher, a value of n is suitable if the number of oracle-queries of the generic player is higher than the cost of the shortcut player. By previous observations, given a set of n tuples, the probability that at least one common key exists for which the property on the ciphertexts is satisfied is $2^{-94n+128}$. Thus, the idea is to estimate the number of (plaintext, ciphertext) pairs that this player has to generate in order to win the game (that is, in order to find with high probability n tuples with the required property). If this number is higher than $2^{44} \cdot n$ for a fixed n, then the other player wins the game.

Since each set of D_a contains 2^{32} different plaintexts, it is possible to construct approximately 2^{63} different couples $\{(p^1, c^1), (p^2, c^2)\}$. Given t different sets of D_a, it is possible to construct $s = 2^{63} \cdot t$ different couples. It follows that one can construct approximately

$$\binom{s}{n} \approx \frac{s^n}{n!}$$

different sets of n different tuples (i.e. n different couples $\{(p^1, c^1), (p^2, c^2)\}$), where the approximation holds for $n \ll s$. Since the probability that a set of n tuples satisfy the above defined relation $\mathcal{R}$ is $2^{-94n+128}$, the generic player must consider at least s different couples such that $s^n/n! \simeq 2^{94n-128}$ or equivalently

$$s \simeq 2^{94-\frac{128}{n}} \cdot (n!)^{\frac{1}{n}}.$$

By this formula, for $n = 8$ this player has to consider approximately $2^{79.9}$ different tuples, or equivalently $2^{48.9}$ (plaintext, ciphertext) pairs (that is, $2^{16.9}$ initial different sets D_j). Indeed, given $2^{16.9}$ initial different cosets of D_a, it is possible to construct approximately $2^{16.9} \cdot 2^{63} = 2^{79.9}$ different couples, that is approximately 2^{624} different sets of 8 tuples. Since each of these sets satisfies the required properties with probability $2^{-94 \cdot 8+128} = 2^{-624}$, he has a good probability to find 8 different tuples with the required property. The cost to generate these $2^{48.9}$ (plaintexts, ciphertexts) pairs is of $2^{48.9}$ oracle-queries (with the assumption 1 oracle-query $\simeq$ 1 encryption). On the other hand, the cost to generate these 8 tuples for the shortcut player is of $8 \cdot 2^{44} = 2^{47}$ (which is smaller). We emphasize that the cost of the generic player is higher than the cost of the shortcut player is satisfied for any value n with $n \geq 8$. In order to make the advantage of the shortcut player more significant, we have chosen an (arbitrary) value of $n = 64$, which implies a cost for the shortcut player of 2^{50} computations and of $2^{65.6}$ computations for the generic player.

8.4.2. 12-round Distinguishers

Using a similar strategy, 12-round distinguishers can be set up for AES. This distinguisher is obtained by extending the previous 10-round distinguishers both at the end and at the beginning. We highlight that this is the *first known-key distinguisher for full AES-192* (and on 12 rounds of AES-128, i.e.

full AES-128 with two more rounds[9]) and it also provides a counterexample to the claims made in [Gil14].

Truncated-Differential. In the know-key distinguisher scenario, the players have to send to the verifier $n \geq 2^{38}$ different tuples of (plaintext, ciphertext) pairs, that is $\{(p_i^1, c_i^1), (p_i^2, c_i^2)\}$ for $i = 0, ..., n-1$, with the following properties:

1. there exist keys k^0, k^1 s.t. for each tuple there exists j for which the two plaintexts belong to the same coset of $\mathcal{D}_j$ after two rounds, that is
$$\exists\, k^0, k^1 \text{ s.t. } \forall i = 0, ..., n-1 \quad \exists j \in \{0, ..., 3\} \text{ s.t. } R^2_{k^0,k^1}(p_i^1) \oplus R^2_{k^0,k^1}(p_i^2) \in \mathcal{D}_j;$$

2. there exist keys k^{11}, k^{12} s.t. for each tuple there exists l for which the two ciphertexts belong to the same coset of $\mathcal{M}_l$ two rounds before, that is
$$\exists k^{11}, k^{12} \text{ s.t. } \forall i = 0, ..., n-1 \; \exists l \in \{0, ..., 3\} \text{ s.t. } R^{-2}_{k^{11},k^{12}}(c_i^1) \oplus R^{-2}_{k^{11},k^{12}}(c_i^2) \in \mathcal{M}_l;$$

where $R^2_{k^0,k^1}(\cdot) \equiv R_{k^1}(R_{k^0}(\cdot))$ and $R^{-2}_{k^{11},k^{12}}(\cdot) \equiv R^{-1}_{k^{11}}(R^{-1}_{k^{12}}(\cdot))$.

A complete analysis of this distinguisher – similar to the one just given for the 10-round one – is proposed in [GR17].

Uniform Distribution. In the known-key distinguisher scenario, the players have to send to the verifier $n \geq 2$ *different* sets of 2^{64} (plaintext, ciphertext) pairs, that is (p_i^j, c_i^j) for $i = 0, ..., 2^{64}-1$ and $j = 0, ..., n-1$, with the following properties:

1. there exist keys k^0, k^1 such that for all $j = 0, ..., n-1$ the texts $\{R_{k^1}(R_{k^0}(p_i^j))\}_i$ are uniformly distributed among the cosets of $\mathcal{D}_I$ for each $I \subseteq \{0, 1, 2, 3\}$ with $|I| = 3$, or equivalently such that for all $j = 0, ..., n-1$ the bytes of the texts $\{R_{k^1}(R_{k^0}(p_i^j))\}_i$ are uniformly distributed;

2. there exist keys k^{11}, k^{12} such that for all $j = 0, ..., n-1$ the texts $\{R^{-1}_{k^{11}}(R^{-1}_{k^{12}}(c_i^j))\}_i$ are uniformly distributed among the cosets of $\mathcal{M}_J$ for each $J \subseteq \{0, 1, 2, 3\}$ with $|J| = 3$, or equivalently such that for all $j = 0, ..., n-1$ the bytes of the texts $\{MC^{-1} \circ R^{-1}_{k^{11}}(R^{-1}_{k^{12}}(c_i^j))\}_i$ are uniformly distributed.

As for the 8- and the 10-round cases, in order to provide evidence that the previous distinguisher is meaningful and using a similar argumentation to the ones proposed in [Gil14], we show that the probability of the generic player to win the game given $n \geq 2$ sets of 2^{64} texts is negligible.

To do this, we claim that this probability is upper bounded by the probability of the following "related" game. Assume $n = 2$ and consider 2 sets of $2^{64} - d$ (plaintext, ciphertext) pairs for $d \geq 5$, that is (p^i, c^i) for each $i = 0, ..., 2^{64} - d - 1$, with the following property: there is a set of keys k^0, k^1 and k^{11}, k^{12} - which can correspond to the set of the entire keys - such that for each one of the two sets, the bytes of $R_{k^1} \circ R_{k^0}(p^i)$ and of $MC^{-1} \circ R^{-1}_{k^{11}} \circ R^{-1}_{k^{12}}(c^i)$ (that is 2-round encryption of p^i and the 2-round decryption of the ciphertexts) are "almost uniform" w.r.t. the definition given before. The goal of the player is to find $2 \cdot d$ texts such that - for each one of the two sets - the bytes of the 2^{64} texts of each set after 2-round encryption/decryption are uniformly distributed. Since this probability is upper bounded by 2^{-25} - see in the following - and since this second game is "related" to the original one, the conclusion follows immediately.

More formally, using the same argumentation proposed by Gilbert (see Prop. 5 of [Gil14]), we prove the following statement.

[9] We emphasize that the AES-128 (in particular, its key schedule) is well defined and can be extended to 12 rounds.

Proposition 21. *For any oracle algorithm $\mathcal{A}$ that makes $\leq N = 2 \cdot 2^{64} = 2^{65}$ oracle queries to a perfect random permutation Π or Π^{-1} of $\{0,1\}^{128}$, the probability that $\mathcal{A}$ outputs $n \geq 2$ sets of 2^{64}-tuple (X_i, Y_i) for $i = 0, ..., 2^{64} - 1$ of Π that satisfies $Y_i = \Pi(X_i)$ and also satisfies $\mathcal{R}$ defined previously is upper bounded by $\binom{10}{5} \times 2^{512} \times \left(\frac{5^{16}}{2^{128} - (2^{64} - 5)}\right)^6 \approx 2^{-25}$.*

More details about this distinguisher and the proof of this proposition can be found in [GR17].

8.4.3. On the Validity of Gilbert's Known-Key Distinguisher

In [GR17], we showed that Gilbert's known-key distinguisher model can lead to results on more rounds than previous expected. Even though the core distinguisher remains at 8 rounds, 12 instead of 10 rounds are achieved. This may raise the question: *Is it possible to extend this distinguisher to 14-round AES?*

The main criticism in order to extend a known-key distinguisher both at the end and at the beginning as in the Gilbert model regards the computational cost to verify the existence of keys such that the n tuples of (plaintexts, ciphertexts) pairs satisfy the relation $\mathcal{R}$. Thus, to success in this task, the main problem is related to set up efficient key-recovery attack that can be used in the Gilbert's model, rather than looking for new properties - which are independent of the key - of AES.

On the other hand, even if Gilbert's Known-Key Distinguisher leads to statements on more rounds of AES than ever before (without related keys) that seem meaningful, it is not clear if such statements can become useful in the sense to e.g. have an impact of hash function use-cases of block ciphers. This has also been noticed in [Gil14], where it is pointed out that even if the strategy proposed by Gilbert allows to set up efficient known-key distinguishers, its "*impact on the security of [...] AES when used as a known key primitive, e.g. in a hash function construction, is questionable*" (see abstract of [Gil14]). Finally, the validity of Gilbert's model was supported by the fact that a total of two extension rounds seem to be the limit in the known-key model, and that likely only a distinguisher that exploits the uniform distribution property can be extended in such way.

For all these reasons and since we disprove both conjectures, we propose - *with more confidence than would could have been possible without our results* - a (new) definition of known-key distinguisher model that rules out Gilbert's and our attacks proposed in this paper. As our results show, this seems necessary for better capturing the original idea of known-key distinguishers as something "between secret-key model and hash function use-cases", where *the known-key model restores its original intent in which the role of the verifier gets back to being marginal.* In more details, our proposal is to distinguish "classical" known-key distinguisher *where the verifier can directly verify the relation $\mathcal{R}$ on the plaintexts and ciphertexts without guessing any key material*, and the "Gilbert" known-key one. Informally, this can be achieved by requiring that the relation $\mathcal{R}$ does not involve any operation that defines E (with the only exception of a group addition, usually XOR) and any guessing of key material.

A "New" Model: "Classical" Known-Key Distinguisher

Taking a step back from the concrete results, what we also showed is that the gap between the known-key model and the chosen-key model may be even larger. Among the possibilities to remedy this counter-intuitive situation, we propose to define a new model that better capture the desire to have something "in-between" the chosen-key and the known-key model. *Our proposal is to distinguish "classical" Known-Key distinguisher - where the verifier can directly verify the relation $\mathcal{R}$ on the plaintexts and ciphertexts without guessing any key material - and the "Gilbert" Known-Key distinguisher.* Roughly speaking, a "classical" Known-Key distinguisher should only exploit properties which have no connection with the details of the underlying primitive $E(\cdot)$ and that are independent of the (value of the) key. In particular, note that *every block cipher is vulnerable to a known-key distinguisher which re-use the key.* For instance, consider the following straightforward distinguishability attack. Assume the goal is to distinguish if an oracle is instantiated by a cipher $E_K(\cdot)$ or by an ideal cipher $\Pi(K,\cdot)$, under a known key K. Given a query X, one gets Y (which can be $Y = E_K(\cdot)$ or $Y = \Pi(K,X)$). Since the details of $E_K(\cdot)$ and the key K are known, one can simply compute $Y_0 = E_K(X)$. If $Y_0 = Y$, one can conclude that the oracle is instantiated by $E_K(\cdot)$. Note that another "weakness" of such a distinguisher is that it allows access to the internal primitives $E_K(\cdot)$.

In order to distinguish a "classical" Known-Key distinguisher from a "Gilbert" Known-Key distinguisher, our suggestion is simply to adapt Def. 17 for the first case[10]:

Definition 22 ([GR17])**.** *Let $E : (K,X) \in \{0,1\}^k \times \{0,1\}^n \to E_K(X) \in \{0,1\}^n$ denote a block cipher of block size n bits. A "classical" known-key distinguisher $(\mathcal{R},\mathcal{A})$ of order $N \geq 1$ consists of (1) a relation $\mathcal{R}$ over the N-tuples of n-bit blocks (2) an algorithm $\mathcal{A}$ that on input a k-bit key K produces in time $T_{\mathcal{A}}$, i.e. in time equivalent with $T_{\mathcal{A}}$ computations of E, an N-tuple $\mathcal{X} = (X_i)$ $i = 1,...,N$ of plaintext blocks and an N-tuple $\mathcal{Y} = (Y_i)$ $i = 1,...,N$ of ciphertext blocks related by $Y_i = E_K(X_i)$ and by $X \mathcal{R} Y$.*

The following conditions must be met:

- *The relation $\mathcal{R}$ has no "(obvious) connection" with the specification of the cipher $E(\cdot)$ (e.g. such relation should not "extensively" re-use the operations – especially, the non-linear ones – that define E) and it is independent of the value of the key;*
- *The relation $\mathcal{R}$ must be $T_{\mathcal{A}}$-intractable relatively to E;*
- *The validity of $\mathcal{R}$ must be efficiently checkable.*

It follows that - due to the first condition on the relation $\mathcal{R}$ - all the "classical" known-key distinguishers present in the literature satisfy the previous definition, but not the "extended Gilbert distinguishers". The problem to formalize - with a proper mathematical definition - the fact "the relation $\mathcal{R}$ has no 'obvious connection' with the specification of the cipher $E(\cdot)$" is open for future research.

8.5. Chosen-Key Distinguisher

As we have already recalled, the goal of an open-key distinguisher is to differentiate between a block cipher E which allows to generate plaintext/ciphertext pairs which exhibit a rare relation, even for a small set of keys or a single key, and an ideal cipher Π that does not have such a property. However, this poses a definitional problem as it was shown already in [CGH04] that any concrete implementable cipher (like the AES) can be trivially distinguished from an ideal cipher.

Roughly speaking, with respect to a (weak) known-key distinguisher, in a chosen-key one the adversary does not only know the key, but he is also able to choose it. Anyway, this difference has a huge impact on the definition of a proper model for such case. To better understand this fact, note the following. Since the key can be chosen in advance, it is also possible for the generic player to perform pre-computation that can be then exploited to win the game. In general, this is not a concrete problem in the case of a known-key distinguisher, since in this case he should perform this pre-computation for almost half of the keys[11], which is obviously infeasible. Instead, for the chosen-key case, it is sufficient to perform this operation for only one key. Thus, how is it possible to take this strategy into account in order to compute the overall computations of the two players in a chosen-key distinguisher? To the best of our knowledge, *finding a proper formal definition that*

[10] In the following definition, the main difference with Def. 17 is emphasized in *italic*.

[11] This is necessary in order to have a big chance that the "known key" – picked up at random – is in the set of keys for which he performed the pre-computation.

captures the intuition behind chosen-key distinguishers has been a challenging task for the last fifteen years and is still an open problem.

We do *not* attempt to address this formalization challenge here, but proceed in the way that is custom in the literature to describe chosen-key distinguisher: *(1st)* describe the (rare) property that occurs for a particular *chosen* key, *(2nd)* show that it can be efficiently constructed for the block cipher (usually using an inside-out approach) and finally *(3rd)* argue or prove in some model that any generic method is less efficient or has low success probability.

8.5.1. Chosen-Key Distinguishers for AES

To the best of our knowledge, the only chosen-key distinguisher for AES in the single-key setting is proposed in [DFJ12]. Here, the chosen-key model asks the adversary to find two plaintexts/ciphertexts pairs such that the two plaintexts are equal in 3 diagonals and the two ciphertexts are equal in 3 anti-diagonals (if the final MixColumns is omitted). Equivalently, using the subspace trail notation, the goal is to find $(p^1, c^1 \equiv R^8(p^1))$ and $(p^2, c^2 \equiv R^8(p^2))$ for $p^1 \neq p^2$ such that $p^1 \oplus p^2 \in \mathcal{D}_I$ and $c^1 \oplus c^2 \in \mathcal{M}_J$ for a certain $I, J \subseteq \{0, 1, 2, 3\}$ such that $|I| = |J| = 1$.

This problem is equivalent to the one proposed in [GP10; JNP13] in the known-key scenario. In particular, the main (and only) difference between the known-key and chosen-key distinguishers is related to the freedom to choose the key, and consequently to the computational cost. In more details, due to this freedom, for the 8-round AES-128 case it is possible to find the required pairs of plaintexts/ciphertexts with 2^{24} computations instead of 2^{44}, while the computational cost in the case of an ideal cipher is of 2^{64} in both cases. For completeness, a similar result is proposed for 9-round AES-256.

The chosen-key model has been popularized some years before by [BKN09], since a distinguisher in this model has been extended to a related-key attack on full AES-256. A related distinguisher for 9-round AES-128 has been proposed by Fouque *et al.* at Crypto 2013 [FJP13]. Both the chosen-key distinguisher proposed in these papers are in the related-key setting. Here we briefly recall them, but we emphasize that we do not consider related-keys in this article. In [BKN09], authors show that it is possible to construct a *q-multicollision*[12] on Davies-Meyer compression function using AES-256 in time $q \cdot 2^{67}$, whereas for an ideal cipher it would require on average $q \cdot 2^{\frac{q-1}{q+1}128}$ time complexity. These results show that AES-256 can not model an ideal cipher in theoretical constructions. A similar approach has been exploited in [FJP13] to set up the first chosen-key distinguisher for 9-round AES-128. Here, the chosen-key model asks the adversary to find a pair of keys (k, k') satisfying $k \oplus k' = \delta$ with a *known (fixed)* difference δ, and a pair of messages $(p^1, c^1 \equiv R^9(p^1))$ and $(p^2, c^2 \equiv R^9(p^2))$ conforming to a partially instantiated differential characteristic in the data part.

8.5.2. New Chosen-Key Distinguishers for AES in the Single-Key Setting

In [GLR+18] we presented new chosen-key distinguishers for AES in the single-key setting. In particular, as a major results, we are able to present *the first candidate 10-round chosen-key distinguisher for AES-128 and a 14-round candidate chosen-key distinguisher for AES-256, both in the single-key setting.* All the distinguishers that we present are based on the (practically verified) multiple-of-n property, similar to the one proposed at Eurocrypt 2017 [GRR17] and adapted in the following for the invariant subspace $\mathcal{IS}$ case.

Our results are summarized in Table 8.3. Since all chosen-key distinguishers that we are going to present work in the same way, we limit ourselves to give all the details for the AES-128 case.

The Subspace $\mathcal{X}$. To present the distinguisher, we first introduce the generic subspace $\mathcal{X}$.

[12] A set of two differences and q pairs $\{\Delta K, \Delta P; (P_1, K_1), (P_2, K_2), \ldots, (P_q, K_q)\}$ is called a differential *q-multicollision* for a cipher $E_K(\cdot)$ if $E_{K_i}(P_i) \oplus E_{K_i \oplus \Delta K}(P_i \oplus \Delta P) = E_{K_j}(P_j) \oplus E_{K_j \oplus \Delta K}(P_j \oplus \Delta P)$ for each $i, j = 1, \ldots, q$.

Table 8.3.: *AES Chosen-Key Distinguishers.* The computation cost is the cost to generate N-tuples of plaintexts/ciphertexts. SK denotes a chosen-key distinguisher in the Single-Key setting, while RK denotes a chosen-key distinguisher in the Related-Key setting. Distinguishers proposed in this paper are in bold.

AES	Rounds	Computations	Property	SK	RK	Reference
AES-128	8	2^{24}	Multiple Diff. Trail	✓		[DFJ12]
	9	2^{55}	Multi-Collision Diff.		✓	[FJP13]
	10 (full)	$\mathbf{2^{64}}$	**Multiple-of-n**	✓		**[GLR+18]**
AES-192	**11**	$\mathbf{2^{64}}$	**Multiple-of-n**	✓		**[GLR+18]**
AES-256	9	2^{24}	Multiple Diff. Trail	✓		[DFJ12]
	14 (full)	$\mathbf{2^{64}}$	**Multiple-of-n**	✓		**[GLR+18]**
	14 (full)	2^{120}	Multi-Collision Diff.		✓	[BKN09]

Definition 23. *Let I a subset of $\{(0,0),(0,1),\ldots,(3,2),(3,3)\} \equiv \{(i,j)\}_{0\leq i,j\leq 3}$. Let the subspace $\mathcal{X}_I$ be defined as*

$$\mathcal{X}_I = \langle \{e_{i,j}\}_{(i,j)\in I} \rangle \equiv \left\{ \bigoplus_{(i,j)\in I} \alpha_{i,j} \cdot e_{i,j} \,\middle|\, \forall \alpha_{i,j} \in \mathbb{F}_{2^8} \right\}.$$

In other words, $\mathcal{X}_I$ is the set of elements given by linear combinations of $\{e_{i,j}\}_{(i,j)\in I}$, where $e_{i,j} \in \mathbb{F}_{2^8}^{4\times 4}$ has a single 1 in row i and column j.

Proposition 22. *For each $I \subseteq \{(0,0),(0,1),\ldots,(3,2),(3,3)\} \equiv \{(i,j)\}_{0\leq i,j\leq 3}$ and for each $a \in \mathcal{X}_I^{\perp}$, there exists one and only one $b \in \mathcal{Y}_I^{\perp}$ such that*

$$R(\mathcal{X}_I \oplus a) = \mathcal{Y}_I \oplus b$$

where $\mathcal{Y}_I = MC \circ SR(\mathcal{X}_I)$.

The proof of this proposition follows from the fact that a coset of $\mathcal{X}_I$ is mapped into a coset of $\mathcal{X}_I$ after the S-Box operation, that is S-Box$(\mathcal{X}_I \oplus a) = \mathcal{X}_I \oplus$ S-Box$(a) = \mathcal{X}_I \oplus b$. Moreover, observe that for each $I \subset \{(0,0),(0,1),\ldots,(3,2),(3,3)\} \equiv \{(i,j)\}_{0\leq i,j\leq 3}$, there exists $J \subseteq \{(i,j)\}_{0\leq i,j\leq 3}$ such that $SR(\mathcal{X}_I) = \mathcal{X}_J$ (or equivalently $SR^{-1}(\mathcal{X}_I) = \mathcal{X}_J$). As a result, $\{\mathcal{X}_I, MC \circ SR(\mathcal{X}_I)\}$ is a subspace trail of length 1. Note that such subspace trail can not be extended on two rounds for any *generic* $\mathcal{X}_I$, due to the non-linear S-Box operation of the next round (that *can* destroy the linear relations that hold among the bytes).

8.6. The "Simultaneous Multiple-of-n" Property - A 9-round chosen-key distinguisher for AES

In our distinguisher, the chosen-key model asks the adversary to find a set of 2^{64} (plaintexts, ciphertexts), that is $(p^i, c^i \equiv R^9(p^i))$ for $i = 0,\ldots,2^{64}-1$ – where all the plaintexts/ciphertexts are generated by *the same key* – such that the following "*simultaneous multiple-of-n*" property is satisfied:

- *for each $J, I \subseteq \{0,1,2,3\}$, the number of different pairs of ciphertexts that belong to the same coset of $\mathcal{M}_J$ and the number of different pairs of plaintexts that belong to the same coset of $\mathcal{D}_I$ are a multiple of* $128 = 2^7$;

- *for each $J, I \subset \{(0,0),(0,1),\ldots,(3,2),(3,3)\} \equiv \{(i,j)\}_{0\leq i,j\leq 3}$, the number of different pairs of ciphertexts that belong to the same coset of $MC(\mathcal{X}_I)$ and the number of different pairs of plaintexts that belong to the same coset of $\mathcal{X}_J$ are a multiple of* 2, where $\mathcal{X}$ is defined as in 23.

For the follow-up, we remark and highlight that the subspaces $\mathcal{X}$ are independent, in the sense that e.g. the fact that the multiple-of-2 property is satisfied by $\mathcal{X}_I$ and/or $\mathcal{X}_J$ does not imply anything on $\mathcal{X}_{I\cup J}$ and vice-versa. This is due to the fact that given $\mathcal{X}_I$ and $\mathcal{X}_J$, then $\mathcal{X}_I \cup \mathcal{X}_J \subsetneq \mathcal{X}_{I\cup J}$ if $\mathcal{X}_{I\cup J} \neq \mathbb{F}_{2^8}^{4\times 4}$. As a result, any information about the multiple-of-n property on $\mathcal{X}_I, \mathcal{X}_J$ (and so $\mathcal{X}_I \cup \mathcal{X}_J$) is useless to derive information about the multiple-of-n property on $\mathcal{X}_{I\cup J} \setminus (\mathcal{X}_I \cup \mathcal{X}_J)$ – assuming $\mathcal{X}_{I\cup J} \neq \mathbb{F}_{2^8}^{4\times 4}$.

8.6.1. Weak-key "Multiple-of-n" property

The "multiple-of-8" property [GRR17] proposed at Eurocrypt 2017 can be summarized as follows. Given $2^{32\cdot|I|}$ plaintexts in the same coset of a diagonal space $\mathcal{D}_I$, the number of different pairs of ciphertexts after 5-round AES that belong to the same coset of $\mathcal{M}_J$ after 5-round AES is always a multiple of 8 with probability 1. This result can be used to set up a distinguisher, since for a random permutation the same property holds with probability 1/8.

In the case of a weak-key, we are able to extend the previous result up to 6-round AES-128. The obtained results are proposed in the following Theorems.

Theorem 9 ([GLR+18]). *Let $\mathcal{IS}$ and $\mathcal{M}_I$ be the subspaces defined as before for a fixed I with $1 \leq |I| \leq 3$. Assume that the whitening key is a weak-key, that is it belongs to the set K_{weak} as defined in 4.9. Given 2^{64} plaintexts in $\mathcal{IS}$, the number n of different pairs[13] of ciphertexts (c^i, c^j) for $i \neq j$ that belong to the same coset of $\mathcal{M}_I$ (that is $c^i \oplus c^j \in \mathcal{M}_I$) has the following property:*

- *for 5-round AES-128, the number of collisions n is a multiple of 128, that is $\exists n' \in \mathbb{N}$ such that $n = 128 \cdot n'$;*
- *for 6-round AES-128, the number of collisions n is a multiple of 2, that is $\exists n' \in \mathbb{N}$ such that $n = 2 \cdot n'$.*

Since the proof of the previous Theorem is similar to the one provided in [GRR17], we limit ourselves here to highlight the crucial points.

Proof. First of all, note that the invariant subspace $\mathcal{IS}$ is mapped into a coset of $\mathcal{IS}$ after 2-round encryption, and similarly a coset of $\mathcal{M}_I$ is mapped into a coset of $\mathcal{D}_I$ after 2-round decryption, that is

$$\forall k \in K_{\text{weak}}: \qquad \mathcal{IS} \xrightarrow[\text{prob. 1}]{R^2(\cdot)} \mathcal{IS} \oplus a \xrightarrow{R(\cdot) \text{ or } R^2(\cdot)} \mathcal{D}_I \oplus a' \xrightarrow[\text{prob. 1}]{R^2(\cdot)} \mathcal{M}_I \oplus b'$$

Thus, the idea is to focus only on the middle round(s), and to prove the following equivalent result. Given 2^{64} plaintexts in a coset of $\mathcal{IS}$, the number n of different pairs of ciphertexts (c^i, c^j) for $i \neq j$ that belong to the same coset of $\mathcal{D}_I$ (that is $c^i \oplus c^j \in \mathcal{D}_I$) after 1 or 2 round(s) has the following property:

- for 1-round AES, the number of collisions n is a multiple of 128;
- for 2-round AES, the number of collisions n is a multiple of 2.

[13] Two pairs (s,t) and (t,s) are considered to be equivalent.

5-round AES. Given a pair of texts $t^1, t^2 \in \mathcal{IS} \oplus a$, we are going to prove that there exist other pair(s) of texts $s^1, s^2 \in \mathcal{IS} \oplus a$ such that

$$R(t^1) \oplus R(t^2) \in \mathcal{D}_I \qquad \textit{if and only if} \qquad R(s^1) \oplus R(s^2) \in \mathcal{D}_I.$$

where the texts s^1, s^2 are given by a different combination of the generating variables of t^1, t^2.

By definition of $\mathcal{IS}$, let t^1 and t^2 be as

$$t^i = a \oplus \begin{bmatrix} x_0^i & x_4^i & x_0^i & x_4^i \\ x_1^i & x_5^i & x_1^i & x_5^i \\ x_2^i & x_6^i & x_2^i & x_6^i \\ x_3^i & x_7^i & x_3^i & x_7^i \end{bmatrix}, \qquad \text{that is } \; t^i = a \oplus \bigoplus_{j=0}^{7} x_j^i \cdot (e_j \oplus e_{j+8}). \tag{8.4}$$

where $x_{l,j}$ or $x_{l+4\times j}$ denotes the byte in the l-th row and in the j-th column. For simplicity, let $t^i \equiv (x_0^i, x_1^i, x_2^i, x_3^i, x_4^i, x_5^i, x_6^i, x_7^i)$. Consider initially the case in which all the variables are different, that is $x_j^1 \neq x_j^2$ for $j = 0, 1, \ldots, 7$. Let S be the set of pairs of texts $s^1, s^2 \in \mathcal{IS} \oplus a$ defined by swapping some generating variables of t^1 and t^2. In particular, given t^1 and t^2, the set S_{t^1,t^2} contains all 128 pairs of texts (s^1, s^2) for all $I \subseteq \{0, 1, 2, 3, 4, 5, 6, 7\}$ where

$$s^1 = a \oplus \bigoplus_{j=0}^{7} \left\{ \left[\left(x_j^1 \cdot \delta_j(I) \right) \oplus \left(x_j^2 \cdot [1 - \delta_j(I)] \right) \right] \cdot \left(e_j \oplus e_{j+8} \right) \right\}$$
$$s^2 = a \oplus \bigoplus_{j=0}^{7} \left\{ \left[\left(x_j^2 \cdot \delta_j(I) \right) \oplus \left(x_j^1 \cdot [1 - \delta_j(I)] \right) \right] \cdot \left(e_j \oplus e_{j+8} \right) \right\}$$

where the pairs (s^1, s^2) and (s^2, s^1) are considered to be equivalent, and where $\delta_x(A)$ is the Dirac measure defined as

$$\delta_x(A) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

As we are going to show, since

$$\forall (s^1, s^2) \in S_{t^1,t^2} : \qquad R(t^1) \oplus R(t^2) = R(s^1) \oplus R(s^2),$$

it follows that

$$\forall (s^1, s^2) \in S_{t^1,t^2} : \qquad R(t^1) \oplus R(t^2) \in \mathcal{D}_I \quad \text{iff} \quad R(s^1) \oplus R(s^2) \in \mathcal{D}_I.$$

The first equivalence depends on the fact that the S-Box operation works independently on each byte and that the XOR-sum is commutative. To show this fact, we compute the byte in position (0, 0) – analogous for the other cases – of the previous difference

$$\begin{aligned} (R(t^1) \oplus R(t^2))_{0,0} = {} & \text{0x02} \cdot [\text{S-Box}(x_0^1 \oplus a'_{0,0}) \oplus \text{S-Box}(x_0^2 \oplus a'_{0,0})] \oplus \\ & \oplus \text{0x03} \cdot [\text{S-Box}(x_5^1 \oplus a'_{1,1}) \oplus \text{S-Box}(x_5^2 \oplus a'_{1,1})] \oplus [\text{S-Box}(x_2^1 \oplus a'_{2,2}) \oplus \\ & \oplus \text{S-Box}(x_2^2 \oplus a'_{2,2})] \oplus [\text{S-Box}(x_7^1 \oplus a'_{3,3}) \oplus \text{S-Box}(x_7^2 \oplus a'_{3,3})] = (R(s^1) \oplus R(s^2))_{0,0} \end{aligned}$$

where $a'_{i,i}$ for $i = 0, 1, 2, 3$ depends on the initial constant a that defines the coset of $\mathcal{IS}$ and on the secret key. Since each set S_{t^1,t^2} has cardinality 128, in the case in which one focuses on the pairs of texts with different generating variables, it follows that the multiple-of-128 property previously defined holds.

What happens if some variables are equal, e.g. $x_j^1 = x_j^2$ for $j \in J \subseteq \{0, ..., 7\}$ with $|J| \geq 1$? In this case, it is possible to prove that the difference $R(t^1) \oplus R(t^2)$ is independent of $x_j^1 = x_j^2$ for each

$j \in J$ (e.g. consider the difference $(R(t^1) \oplus R(t^2))_{0,0}$ in the byte (0,0) just given). As a result, the idea is to consider all the different pairs of texts given by swapping one or more variables x_l^1 and x_l^2 for $l = 0, 1, \ldots, 7$, where x_j for $j \in J$ can takes any possible value in $\mathbb{F}_{2^8}$. Note that in the case in which $0 \leq |J| < 8$ variables are equal, it is possible to identify

$$\underbrace{2^{7-|J|}}_{\text{by swapping different gen. variables}} \times \underbrace{2^{8\cdot|J|}}_{\text{due to equal gen. variables}} = 2^{7\cdot(1+|J|)} \geq 128$$

different texts s^1 and s^2 in $\mathcal{IS} \oplus a$ that satisfy the condition $R(t^1) \oplus R(t^2) = R(s^1) \oplus R(s^2)$.

In conclusion, given plaintexts in the same coset of $\mathcal{IS}$, the number of different pairs of ciphertexts that belong to the same coset of $\mathcal{D}_I$ after one round is a multiple of 128. More formally, given t^1 and t^2, the set S_{t^1,t^2} contains all $2^{7\cdot(1+|J|)}$ pairs of texts (s^1, s^2) for all $I \subseteq \{0, 1, 2, 3, 4, 5, 6, 7\} \setminus J$ and for all $\alpha_0, ..., \alpha_{|J|} \in \mathbb{F}_{2^8}$ where

$$s^1 = a \oplus \bigoplus_{j \in \{0,\ldots,7\} \setminus J} \left\{ \left[\left(x_j^1 \cdot \delta_j(I) \right) \oplus \left(x_j^2 \cdot [1 - \delta_j(I)] \right) \right] \cdot \left(e_j \oplus e_{j+8} \right) \right\} \oplus \bigoplus_{j \in J} \alpha_j \cdot \left(e_j \oplus e_{j+8} \right)$$

$$s^1 = a \oplus \bigoplus_{j \in \{0,\ldots,7\} \setminus J} \left\{ \left[\left(x_j^2 \cdot \delta_j(I) \right) \oplus \left(x_j^1 \cdot [1 - \delta_j(I)] \right) \right] \cdot \left(e_j \oplus e_{j+8} \right) \right\} \oplus \bigoplus_{j \in J} \alpha_j \cdot \left(e_j \oplus e_{j+8} \right)$$

6-round AES: Super-Sbox. In order to prove the previous claim, we use the "super-Sbox" notation (3.1). Given a pair of texts $t^1, t^2 \in \mathcal{IS} \oplus a$, we prove that there exist other pair(s) of texts $s^1, s^2 \in \mathcal{IS} \oplus a$ such that

$$R^2(t^1) \oplus R^2(t^2) \in \mathcal{D}_I \qquad \textit{if and only if} \qquad R^2(s^1) \oplus R^2(s^2) \in \mathcal{D}_I$$

where the texts s^1, s^2 are obtained by swapping the diagonals of t^1, t^2. In more details, if the columns are different (i.e., $[x_0^1, x_2^1, x_5^1, x_7^1] \neq [x_0^2, x_2^2, x_5^2, x_7^2]$ and $[x_1^1, x_3^1, x_4^1, x_6^1] \neq [x_1^2, x_3^2, x_4^2, x_6^2]$), given t^1 and t^2 defined as in (8.4)

$$SR(t^i) \equiv \Big(\underbrace{[x_0^i, x_2^i, x_5^i, x_7^i]}_{1st \text{ and } 3rd \text{ columns}} \,,\, \underbrace{[x_1^i, x_3^i, x_4^i, x_6^i]}_{2nd \text{ and } 4th \text{ columns}} \Big),$$

then s^1 and s^2 are defined as

$$SR(s^i) \equiv \Big(\underbrace{[x_0^{3-i}, x_2^{3-i}, x_5^{3-i}, x_7^{3-i}],}_{1st \text{ and } 3rd \text{ columns}} \; \underbrace{[x_1^i, x_3^i, x_4^i, x_6^i]}_{2nd \text{ and } 4th \text{ columns}} \Big)$$

and where (s^1, s^2) and (s^2, s^1) are considered to be equivalent.

To prove the previous fact, we first recall that 2-round encryption can be rewritten using the super-Sbox notation

$$R^2(\cdot) = ARK \circ MC \circ SR \circ \textit{super-Sbox} \circ SR(\cdot).$$

Thus, we are going to prove that

$$\textit{super-Sbox}(\hat{t}^1) \oplus \textit{super-Sbox}(\hat{t}^2) \in \mathcal{W}_I \qquad \text{iff} \qquad \textit{super-Sbox}(\hat{s}^1) \oplus \textit{super-Sbox}(\hat{s}^2) \in \mathcal{W}_I$$

where

$$\hat{t}^i = SR(t^i) \in \mathcal{IS} \oplus SR(a) \qquad \text{and} \qquad \hat{s}^i = SR(s^i) \in \mathcal{IS} \oplus SR(a)$$

for $i = 1, 2$ (note that $t^i, s^i \in \mathcal{IS} \oplus a$) and where the subspace $\mathcal{W}_I$ is defined as

$$\mathcal{W}_I = SR^{-1} MC^{-1}(\mathcal{D}_I).$$

Note that the first and the third columns of $\hat{t}^i$ and $\hat{s}^i$ are equal, as well as the second and the fourth columns. Similar to the 5-round case, *since the first and the second columns (and so the third and* *the fourth ones) of $\hat{t}^1$ and $\hat{t}^2$ depend on different and independent variables, since the super-Sbox works independently on each column and since the XOR-sum is commutative,* it follows that

$$\textit{super-Sbox}(\hat{t}^1) \oplus \textit{super-Sbox}(\hat{t}^2) = \textit{super-Sbox}(\hat{s}^1) \oplus \textit{super-Sbox}(\hat{s}^2)$$

which implies the thesis.

What happens if one diagonal is in common for the two texts, e.g. $[x_0^1, x_2^1, x_5^1, x_7^1] = [x_0^2, x_2^2, x_5^2, x_7^2]$ (analogous for $[x_1^1, x_3^1, x_4^1, x_6^1] = [x_1^2, x_3^2, x_4^2, x_6^2]$)? As before, in this case the difference $R^2(t^1) \oplus R^2(t^2)$ is independent of the values of such diagonal. It follows that the pair of texts s^1 and s^2 can be constructed as

$$SR(s^i) \equiv \big(\underbrace{[x_1^{3-i}, x_3^{3-i}, x_4^{3-i}, x_6^{3-i}]}_{1st \text{ and } 3rd \text{ columns}}, \underbrace{[\alpha_0, \alpha_2, \alpha_5, \alpha_7]}_{2nd \text{ and } 4th \text{ columns}}\big) \text{ or } SR(s^i) \equiv \big(\underbrace{[x_1^{i}, x_3^{i}, x_4^{i}, x_6^{i}]}_{1st \text{ and } 3rd \text{ columns}}, \underbrace{[\alpha_0, \alpha_2, \alpha_5, \alpha_7]}_{2nd \text{ and } 4th \text{ columns}}\big)$$

where $\alpha_0, \alpha_2, \alpha_5, \alpha_7$ can take any possible values in $\mathbb{F}_{2^8}$. Note that in the case, it is possible to identify $2 \cdot 2^{32} = 2^{33} \geq 2$ different texts s^1 and s^2 in $\mathcal{IS} \oplus a$ that satisfy the condition $R^2(t^1) \oplus R^2(t^2) = R^2(s^1) \oplus R^2(s^2)$. □

In a similar way, it is possible to prove the following Theorem.

Theorem 10 ([GLR+18]). *Let $\mathcal{IS}$ and $\mathcal{X}_I$ be the subspaces defined as before, for an arbitrary $I \subset \{(0,0),(0,1),\ldots,(3,2),(3,3)\} \equiv \{(i,j)\}_{0\leq i,j \leq 3}$. Assume that the whitening key is a weak-key, i.e. it belongs to the set K_{weak} defined in 4.9. Given 2^{64} plaintexts in $\mathcal{IS}$, the number n of different pairs of ciphertexts (c^i, c^j) for $i \neq j$ that belong to the same coset of $\mathcal{X}_I$ (i.e. $c^i \oplus c^j \in \mathcal{X}_I$) has the following property independently of the details of the S-Box:*

- *for 5-round AES-128, the number of collisions n is a multiple of 2, that is $\exists n' \in \mathbb{N}$ such that $n = 2 \cdot n'$.*

To prove this result, note that with probability 1

$$\forall k \in K_{\text{weak}}: \qquad \mathcal{IS} \xrightarrow[\text{prob. 1}]{R^2(\cdot)} \mathcal{IS} \oplus a \xrightarrow{R^2(\cdot)} \mathcal{Y}_I \oplus a' \xrightarrow[\text{prob. 1}]{R(\cdot)} \mathcal{X}_I \oplus b'$$

where $\mathcal{Y}_I = SR^{-1} \circ MC^{-1}(\mathcal{X}_I)$, as showed in (22) (remember that S-Box$(\mathcal{X}_I) = \mathcal{X}_I$).

Using the same technique as before (i.e. working with the *super-Sbox* notation and by swapping the generating diagonals of a pair of texts), the idea is to focus on the middle rounds only, and to show that given 2^{64} plaintexts in a coset of $\mathcal{IS}$, the number n of different pairs of ciphertexts (c^i, c^j) for $i \neq j$ that belong to the same coset of $\mathcal{Y}_I$ (that is $c^i \oplus c^j \in \mathcal{Y}_I$) after 2 rounds is always a multiple of 2.

Multiple-of-n Property for AES-192/256. As we have seen in Sect. 4.4.3, it is possible to set up a weak invariant subspace of length two for 2^{64} weak-keys of AES-192, and a weak invariant subspace of length two/four/five for $2^{128}/2^{64}/2^{32}$ weak-keys of AES-256. Due to the argumentation just given, it follows that the multiple-of-128 property holds for up to 7-round AES-256, while the multiple-of-2 property holds for up to 9-round AES-256.

8.6.2. 9-round Chosen-Key Distinguisher for AES-128

To find a set of 2^{64} plaintexts/ciphertexts with the required "simultaneous multiple-of-n" property, the distinguisher exploits the fact that *the required property can be fulfilled by starting in the middle with a suitable set of texts.* In particular, the idea is simply to *choose the key such that the subkey of the 4-th round k^4 belongs the subset K_{weak} defined as in 4.9.* Thus, consider the invariant subspace $\mathcal{IS}$ defined as in 4.7, and define the 2^{64} plaintexts as the 4-round decryption of $\mathcal{IS}$ and the corresponding ciphertexts as the 5-round encryption of $\mathcal{IS}$. Due to the secret-key distinguishers just presented, this set satisfies the required "simultaneous multiple-of-n" property.

In more details, due to the assumption on the key (that is, $k^4 \in K_{weak} \subseteq \mathcal{IS}$), note that the subspace $\mathcal{IS}$ is mapped into a coset of $\mathcal{IS}$ after two rounds encryption and one round decryption, that is

$$\forall k^4 \in K_{weak}: \qquad \mathcal{IS} \oplus \hat{k} \xleftarrow{R^{-1}(\cdot)} \mathcal{IS} \xrightarrow{R^2(\cdot)} \mathcal{IS} \oplus \tilde{k}.$$

Due to the results of 8.6.1, the multiple-of-n properties hold with probability 1 on the plaintexts and on the ciphertexts

$$\textit{Multiple-of-n} \xleftarrow{R^{-3}(\cdot)} \mathcal{IS} \oplus \hat{k} \xleftarrow{R^{-1}(\cdot)} \mathcal{IS} \xrightarrow{R^2(\cdot)} \mathcal{IS} \oplus \tilde{k} \xrightarrow{R^3(\cdot)} \textit{Multiple-of-n}$$

It follows that the required set can be constructed using 2^{64} computations. Moreover, we emphasize that our experiments on the secret-key distinguishers of 8.6.1 implies the *practical verification of this distinguisher*. What remains is to give arguments as to why producing that property simultaneously on the plaintext and ciphertext side of an ideal cipher is unlikely to be as efficient.

8.6.3. Achieving the "Simultaneous Multiple-of-n" Property Generically

In this case, the adversary faces a family of random and independent *ideal ciphers* $\{\Pi(K, \cdot), K \in \{0,1\}^k\}$, where $k = 128, 192, 256$ respectively for the cases AES-128/192/256. His goal is to find a key k and a set of 2^{64} plaintexts/ciphertexts $(p^i, c^i = \Pi(k, c^i))$ such that the "simultaneous multiple-of-n" property is satisfied. As we are going to show, *the probability to find a set of* 2^{64} *plaintexts/ciphertexts pairs* (X_i, Y_i) *that satisfies the "simultaneous multiple-of-n" property for a random permutation is upper bounded by* $2^{-65\,618}$.

As first thing, we discuss the freedom to choose the key. Since the adversary does not know the details of the ideal cipher Π, he does not have any advantage to choose a particular key instead of another one. For this reason, in the following we limit ourselves to consider the case in which the permutation Π is instantiated by a key chosen at random in the set $\{0,1\}^k$.

Our goal is to prove that the success probability of any oracle algorithm of overall time complexity upper bounded by 2^{64} is negligible.

Proposition 23 ([GLR+18]). *Given an ideal cipher* Π *or* Π^{-1} *of* $\{0,1\}^{128}$ *instantiated by a fixed key uniformly chosen at random in* $\{0,1\}^k$*, consider* $N = 2^{64}$ *oracle queries made by any algorithm* $\mathcal{A}$ *to the ideal cipher* Π *or* Π^{-1}*. Denote this set of* 2^{64} *plaintexts/ciphertexts pairs by* (X_i, Y_i) *for* $i = 0, \dots, 2^{64}-1$*, where* $Y_i = \Pi(X_i)$*. The probability that* $\mathcal{A}$ *outputs a set of* 2^{64} *plaintexts/ciphertexts pairs* (X_i, Y_i) *for* $i = 0, \dots, 2^{64}-1$ *that satisfies the "simultaneous multiple-of-n" property is upper bounded by* $2^{-65\,618}$*.*

Proof. For completeness, consider first the case of a dishonest algorithm $\mathcal{A}$. Given $N-1$ pairs (X_i, Y_i) generated by the ideal cipher Π or Π^{-1}, assume the player chooses X_N and Y_N in order to satisfy the "simultaneous multiple-of-n" property. If at least one of the N pairs (X_i, Y_i) output by $\mathcal{A}$ does not result from a query X_i to Π or a query Y_i to Π^{-1}, then the probability that for this pair $Y_i = \Pi(X_i)$ and consequently the success probability of $\mathcal{A}$ is upper bounded[14] by $\frac{1}{2^{128}-(N-1)}$.

From now one, we consider only the case of honest algorithm $\mathcal{A}$, that is we assume all the pairs (X_i, Y_i) result from queries to Π or Π^{-1}. Consider a (random) set of $2^{64}-1$ plaintexts/ciphertexts pairs $\{(X_i, Y_i)\}_{i=0,\dots,2^{64}-2}$ such that there exists (at least) one plaintext/ciphertext pair $(\hat{X}, \hat{Y})$ for which the required multiple-of-n property is satisfied. By assumption, the player can always find $\hat{X}'$ (resp. $\hat{Y}'$) such that the "simultaneous multiple-of-n" property is satisfied for the plaintexts (resp. for the ciphertexts). However, the oracle answer $\hat{Y}'$ (resp. $\hat{X}'$) is *uniformly drawn* from

[14] Note that there are 2^{128} different pairs (X, Y). If $N-1$ are already given, the probability that $Y_i = \Pi(X_i)$ holds is $(2^{128} - (N-1))^{-1}$.

$\{0,1\}^{128} \setminus \{Y_1, Y_2, \ldots, Y_{2^{64}-1}\}$ (resp. from $\{0,1\}^{128} \setminus \{X_1, X_2, \ldots, X_{2^{64}-1}\}$). Therefore, *the probability that the answer to the N-th query allows the output of $\mathcal{A}$ to satisfy property $\mathcal{R}$ (i.e. multiple-of-n) is upper bounded by* $(2^{-1})^{2^{16}-16} \cdot (2^{-7})^{14} = 2^{-65\,618} \simeq 2^{-2^{16}}$ since

- *there are $\sum_{i=1}^{15} \binom{16}{i} = 2^{16} - 2$ different subspaces $\mathcal{X}_I$ for which the multiple-of-2 property holds, and among them there are 14 subspaces $\mathcal{M}_I$ for which the multiple-of-128 property holds*;
- the probability that the number of collisions is a multiple of N is (approximately) $1/N$.

In order to prove this second point, we first show that the probabilistic distribution of the number of collisions is a binomial distribution[15].

Given a set of n pairs texts, consider the event that m pairs belong to the same coset of a subspace $\mathcal{X}$. As first thing, the probabilistic distribution of number of collisions is simply described by a *binomial distribution*. By definition, a binomial distribution with parameters n and p is the discrete probability distribution of the number of successes in a sequence of n independent yes/no experiments, each of which yields success with probability p. In our case, given n pairs of texts, each of them satisfies or not the above property/requirement with a certain probability. Thus, this model can be described using a binomial distribution, for which the mean μ and the variance σ^2 are respectively given by $\mu = n \cdot p$ and $\sigma^2 = n \cdot p \cdot (1-p)$.

In our case, the number of pairs is given by $\binom{2^{64}}{2} \simeq 2^{127}$, the probability that a pair of texts belong to the same coset of $\mathcal{X}_I$ is equal to $2^{-8\cdot(16-|I|)}$, while it is equal to $2^{-32\cdot(4-|J|)}$ for the subspaces $\mathcal{D}_J$ and $\mathcal{M}_J$.

Probability that "the number of collision is even" is (approximately) $1/2$ *– Case: subspaces* $\mathcal{X}_I$. The probability that the number of collisions is even is given by

$$\sum_{k=0}^{n/2} \binom{n}{2k} \cdot p^{2k} \cdot (1-p)^{n-2k} = \frac{1}{2} + \frac{1}{2} \cdot (1-2p)^n$$

where note that n is an even number. In our case, since $n \simeq 2^{127}$ and $2^{-120} \le p \le 2^{-8}$ (where the prob. 2^{-120} and 2^{-8} correspond resp. to the cases $|I| = 15$ and $|I| = 1$), the previous probability is well approximated by $1/2 + 1/2 \cdot (1-2^{-7})^{2^{127}} \approx 1/2$.

In order to prove the previous result, let X a binomial distribution $X \sim \mathcal{B}(n,p)$. Combining the facts that

$$Prob(X \text{ even}) + Prob(X \text{ odd}) = \sum_{k=0}^{n} \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k} = [(1-p)+p]^n = 1$$
$$Prob(X \text{ even}) - Prob(X \text{ odd}) = \sum_{k=0}^{n} \binom{n}{k} \cdot (-p)^k \cdot (1-p)^{n-k} = [(1-p)-p]^n$$

where

$$Prob(X \text{ even}) = \sum_{k=0}^{n/2} \binom{n}{2k} \cdot p^{2k} \cdot (1-p)^{n-2k}$$
$$Prob(X \text{ odd}) = \sum_{k=0}^{n/2-1} \binom{n}{2k+1} \cdot p^{2k+1} \cdot (1-p)^{n-2k-1},$$

[15] We highlight that the fact that "the probability that the number of collisions is a multiple of N is $1/N$" is obvious *if* the probabilistic distribution of the number of collisions is a uniform one, which is not the case.

it follows that $Prob(X \text{ even}) = \frac{1}{2} + \frac{1}{2} \cdot (1-2p)^n$.

Probability that "the number of collision is a multiple of N" is (approximately) $1/N$ – Case: subspaces $\mathcal{M}_J$ and $\mathcal{D}_J$. In order to prove this result, we first approximate the binomial distribution with a normal one. De Moivre-Laplace Theorem claims that the normal distribution is a good approximation of the binomial one *if* the skewness of the binomial distribution – given by $(1-2p)/\sqrt{n \cdot p \cdot (1-p)}$ – is close to zero. In our case, since $n \simeq 2^{127}$ and $2^{-96} \le p \le 2^{-32}$ (where the prob. 2^{-96} and 2^{-32} correspond resp. to the cases $|J| = 3$ and $|J| = 1$), it follows that $2^{-47.5} \le skew \le 2^{-15.5}$, which means that the normal approximation is sufficiently good. Thus, we approximate the binomial distribution with a normal one $\mathcal{N}(\mu = n \cdot p, \sigma^2 = n \cdot p \cdot (1-p))$, where the probability density function is given by $\varphi(x) = \frac{1}{\sqrt{2\pi \cdot \sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

In this case, what is the probability that the multiple-of-N collisions is satisfied? To answer this question, it is sufficient to sum all the probabilities where the number of collisions is a multiple-of-N (for $N \in \mathbb{N}$ and $N \neq 0$), that is

$$\sum_{x \in \mathbb{Z}} \frac{1}{\sqrt{2\pi \cdot \sigma^2}} e^{-\frac{(N \cdot x - \mu)^2}{2\sigma^2}} = \frac{1}{N} \cdot \underbrace{\sum_{x \in \mathbb{Z}} \frac{1}{\sqrt{2\pi \cdot \tilde{\sigma}^2}} e^{-\frac{(x-\tilde{\mu})^2}{2\tilde{\sigma}^2}}}_{=1 \text{ by definition}} = \frac{1}{N}$$

where $\tilde{\mu} = \mu/N$ and $\tilde{\sigma}^2 = \sigma^2/N^2$. Obviously, if $N = 1$, then this probability is equal to 1.

□

What happens if the adversary performs more than 2^{64} computations? To answer this question, we first compute the probability that a random set of 2^{64} plaintexts/ciphertexts generated by the same key satisfies the "simultaneous multiple-of-n" property. As we have just seen, the "simultaneous multiple-of-n" property is satisfied with probability $(2^{-65\,618})^2 = 2^{-131\,236} \simeq 2^{-2^{17.002}}$.

As a result, given $2^{64} + 2^{12}$ random texts, the player can find a set of 2^{64} texts that satisfy the required property both on the plaintexts and on the ciphertexts, since it is possible to construct

$$\binom{2^{64} + 2^{12}}{2^{64}} \approx \frac{(2^{64})^{2^{12}}}{2^{12}!} \simeq 2^{2^{17.7}}$$

different sets of 2^{64} texts (where $n! \simeq (n/e)^n \cdot \sqrt{2\pi n}$ by Stirling's approximation).

On the other hand, *the cost to identify the right 2^{64} texts among all the others is in general much higher than 2^{64} computations.* Indeed, to have a chance of success higher than 95%, one must consider approximately $3 \cdot 2^{131\,236}$ different sets, since $1 - (1 - 2^{-131\,236})^{3 \cdot 2^{131\,236}} \simeq 1 - e^{-3} \equiv 0.95$, which implies an overall cost much higher than the cost of the distinguisher.

Moreover, consider the following. Given a set of random texts, suppose to change one plaintext in order to modify the number of collisions in the subspace $\mathcal{X}_I$ (or/and $\mathcal{D}_I$) for a particular I. The problem is that all the other numbers of collisions in the subspace $\mathcal{X}_J$ (or/and $\mathcal{D}_J$) for all $J \neq I$ change. Even if it is possible to have control of these numbers, also the numbers of collisions among the ciphertexts in each subspace $MC(\mathcal{X}_K)$ and $\mathcal{M}_K$ change, and in general it is not possible to predict such change in advance. In particular, we recall that the number of collisions in a subspace $\mathcal{D}_I$ (resp. $\mathcal{M}_I$) is on average $2^{127} \cdot 2^{-128+32 \cdot |I|} = 2^{32 \cdot |I| - 1} \gg 1$, which implies that the change in one text modifies all the numbers of collisions in each subspaces $\mathcal{D}_I$ or/and $\mathcal{M}_I$ for each $I \subseteq \{0, 1, 2, 3\}$. Similarly, the number of pairs of texts with $1 \le |J| \le 15$ equal bytes (that is, that belong to the same coset of a particular subspace $\mathcal{X}_J$) is on average equal to $2^{127} \cdot 2^{-8 \cdot |J|} \ge 2^{127} \cdot 2^{-8 \cdot 15} = 2^7$, which implies that the change in one text modifies all the numbers of collisions in each subspaces $\mathcal{X}_J$ or $MC(\mathcal{X}_J)$ for each $J \subseteq \{e_{i,j}\}_{0 \le i,j \le 3}$. We conjecture that that *there is no (efficient) strategy* – that does not involve brute force research – *to fulfill the required "simultaneous multiple-of-n" property* for which the cost is approximately of 2^{64} computations (or lower). The problem to *formally* prove this fact is left for future work.

Remark. Before going on, we highlight that this claim/result is not true in general if one considers *only* a multiple-of-n property only in subspaces $\mathcal{D}_I$ and $\mathcal{M}_J$ (that is, not generic subspace $\mathcal{X}$) for $n < 8$. In particular, in [GLR+18, App. F] we consider a distinguisher on full AES-192 which is based on the simultaneous multiple-of-2 property both on the plaintexts (in $\mathcal{D}_I$) and the ciphertexts (in $\mathcal{M}_J$). In that section, we present a strategy that the adversary can use to win the game at (almost) the same cost of the distinguisher.

8.6.4. Chosen-key distinguisher for 10-round AES-128

To set up the chosen-key distinguisher for 10-round AES-128, two possible approaches can be considered:

- use the previous distinguisher on 9-round as a starting point and *add one round in the middle* by using the remaining degrees of freedom in the choice of the key;
- use the previous distinguisher on 9-round as a starting point and *add one round at the beginning (or at the end)* by exploiting a weaker property on the plaintexts (or on the ciphertexts).

One (more) Round in the Middle. As we have seen, if the subkey k^4 of the 4-th round belongs in K_{weak} (defined as in 4.9), it follows that

$$\textit{Multiple-of-n} \xleftarrow{R^{-3}(\cdot)} \mathcal{IS} \oplus a \xleftarrow{R^{-1}(\cdot)} \mathcal{IS} \xrightarrow{R^{2}(\cdot)} \mathcal{IS} \oplus b \xrightarrow{R^{3}(\cdot)} \textit{Multiple-of-n}$$

In other words, one exploits the fact that the subspace $\mathcal{IS}$ is mapped into a coset of it after 2-round encryption and 1-round decryption for any subkey in K_{weak}.

By simple computation, there is a key in K_{weak} for which the subspace $\mathcal{IS}$ is mapped into one of its coset after two rounds decryption. In more details, for the key $\hat{k} \in K_{weak}$ defined by

$$\hat{k} \equiv (A = 0\text{x}63 \oplus R[5], B = 0\text{x}63, C = 0\text{x}63, D = 0\text{x}63) \in K_{weak}$$

it follows that

$$\mathcal{IS} \oplus a \xleftarrow{R^{-2}(\cdot)} \mathcal{IS} \xrightarrow{R^{2}(\cdot)} \mathcal{IS} \oplus b.$$

To see this, it is sufficient to compute one round of the key schedule

$$\begin{bmatrix} A \oplus 0\text{x}63 \oplus R[5] & 0 & 0 & 0 \\ B \oplus 0\text{x}63 & 0 & 0 & 0 \\ C \oplus 0\text{x}63 & 0 & 0 & 0 \\ D \oplus 0\text{x}63 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{\text{1-round } \textit{key-schedule}} K_{weak} \equiv \begin{bmatrix} A & A & A & A \\ B & B & B & B \\ C & C & C & C \\ D & D & D & D \end{bmatrix},$$

and to look for a key in K_{weak} that belongs to $\mathcal{IS}$ one round before. As a result, it follows that for the key $\hat{k} \equiv (A = 0\text{x}63 \oplus R[5], B = 0\text{x}63, C = 0\text{x}63, D = 0\text{x}63) \in K_{weak}$ it is possible to set up a distinguisher on 10 rounds[16] since

$$\textit{Multiple-of-n} \xleftarrow{R^{-3}(\cdot)} \mathcal{IS} \oplus a \xleftarrow{R^{-2}(\cdot)} \mathcal{IS} \xrightarrow{R^{2}(\cdot)} \mathcal{IS} \oplus b \xrightarrow{R^{3}(\cdot)} \textit{Multiple-of-n}$$

Using this observation, we can construct the distinguisher. Exactly as before, the chosen-key model asks the adversary to find a set of 2^{64} plaintexts/ciphertexts, i.e. $(p^i, c^i \equiv R^{10}(p^i))$ for $i = 0, \ldots, 2^{64}-1$ – where all the plaintexts/ciphertexts are generated by *the same key* – such that

[16] For completeness, we discuss the relevance of a distinguisher that can be constructed for a single key (which this does not mean – in general – that it holds for one key only). A single collision/near-collision/ or similar distinguishing property for a block-cipher based compression function or hash function would be also a property of the cipher that holds (depending on the mode) for a single key. Assume this is found with a non-generic approach. This simple example shows that, in principle, properties even for single keys can be interesting.

- for each $J, I \subseteq \{0,1,2,3\}$, the number of different pairs of ciphertexts that belong tkeccak day o the same coset of $\mathcal{M}_J$ and the number of different pairs of plaintexts that belong to the same coset of $\mathcal{D}_I$ are a multiple of $128 \equiv 2^7$;
- for each $J, I \subset \{(0,0),(0,1),\ldots,(3,2),(3,3)\} \equiv \{(i,j)\}_{0\le i,j\le 3}$, the number of different pairs of ciphertexts that belong to the same coset of $MC(\mathcal{X}_I)$ and the number of different pairs of plaintexts that belong to the same coset of $\mathcal{X}_J$ are a multiple of 2.

Same as for the 9-round case, due to our argumentations from 8.6.3 we conjecture that the computational cost of an adversary to generate such set is (much) higher than 2^{64} computations.

A Weaker Property. In this case, the chosen-key model asks the adversary to find a set of 2^{64} (plaintexts, ciphertexts), that is $(p^i, c^i \equiv R^{10}(p^i))$ for $i = 0, \ldots, 2^{64}-1$ – where all the plaintexts/ciphertexts are generated by *the same key* – such that the following “*simultaneous multiple-of-n*” property is satisfied:

Plaintext: on the plaintexts, we re-use the previous properties: *(1st)* for each $J \subseteq \{0,1,2,3\}$, the number of different pairs of plaintexts that belong to the same coset of $\mathcal{D}_J$ is a multiple of $128 = 2^7$; *(2nd)* for each $I \subset \{(0,0),(0,1),\ldots,(3,2),(3,3)\} \equiv \{(i,j)\}_{0\le i,j\le 3}$, the number of different pairs of plaintexts that belong to the same coset of $\mathcal{X}_I$ are a multiple of 2;

Ciphertext: *for each* $J \subseteq \{0,1,2,3\}$*, the number of different pairs of ciphertexts that belong to the same coset of* $\mathcal{M}_J$ *is a multiple of* 2.

Choosing one of the 2^{32} keys proposed for the 9-round distinguisher given in 8.6.2, it is possible to construct such set with a computational cost of 2^{64}. In more details, due to the assumption on the key (that is, $k^4 \in K_{weak} \subseteq \mathcal{IS}$), note that the subspace $\mathcal{IS}$ is mapped into a coset of $\mathcal{IS}$ after two rounds encryption and one round decryption, that is

$$\forall k^4 \in K_{weak}: \qquad \mathcal{IS} \oplus \hat{k} \xleftarrow{R^{-1}(\cdot)} \mathcal{IS} \xrightarrow{R^2(\cdot)} \mathcal{IS} \oplus \tilde{k}.$$

Due to the results of 8.6.1, the multiple-of-128 property (on $\mathcal{D}_J$) and the multiple-of-2 property (on $\mathcal{X}_I$) hold with probability 1 on the plaintexts while the multiple-of-2 property (on $\mathcal{M}_J$) holds on the ciphertexts

$$\textit{Multiple-of-n} \xleftarrow{R^{-3}(\cdot)} \mathcal{IS} \oplus \hat{k} \xleftarrow{R^{-1}(\cdot)} \mathcal{IS} \xrightarrow{R^2(\cdot)} \mathcal{IS} \oplus \tilde{k} \xrightarrow{R^4(\cdot)} \textit{Multiple-of-}2$$

What about an adversary facing a family of random and independent ideal ciphers*?* Due to previous analysis, the property on the plaintexts is satisfied with prob. $2^{-32\,809} \simeq 2^{-2^{15}}$ while the property on the ciphertexts is satisfied with prob. 2^{-14}, for an overall probability of $2^{-32\,809} \cdot 2^{-14} = 2^{-32\,823} \simeq 2^{-2^{15}}$.

In other words, the property on the ciphertexts is much weaker than the property on the plaintexts. This fact can be potentially used to generate a set of 2^{64} plaintexts/ciphertexts with the required properties with a data cost of $3 \cdot 2^{78}$. Indeed, the attacker can easily generate a set of 2^{64} plaintexts that satisfy the “Multiple-of-n” property as described before (e.g. he can generate such set using the fact that the 4-round AES decryption of $\mathcal{IS}$ – namely $R^4(\mathcal{IS})$ – has the required “Multiple-of-n” property). Then, he simply asks the oracle for the corresponding ciphertexts, which satisfy the “Multiple-of-2” property with prob. 2^{-14}. By repeating this process $3 \cdot 2^{14}$, the probability of success[17] is higher than 95%. The cost of such strategy (which includes both the generation of the texts and the check that the property is satisfied) is *at least* of 2^{78}.

[17] The probability of success is given by $1-(1-2^{-14})^{3\cdot 2^{14} \geq 0.95}$.

Even if this attack is faster than 2^{128}, its cost is still (much) bigger than 2^{64}, which is the cost to generate the required set of plaintexts/ciphertexts for the case of 10-round AES. Remember that the goal in an open-key distinguisher is indeed to be able to generate the requires set of plaintexts/ciphertexts with a *similar* (or even the same) cost for AES (or the studied cipher) and for the ideal cipher. In this case, it is very unlikely that any generic attack can get close to that: *even if we would allow unlimited time, the data complexity of a generic attack would still need to be higher than* 2^{64}. Indeed, working as in the 9-round case, a simple brute force attack requires at least[18] $2^{64} + 2^{11}$ plaintexts/ciphertexts in order to find a set of 2^{64} plaintexts with the required properties. For all these reasons and same as for the 9-round case (see our arguments from 8.6.3), we conjecture that the data/computational cost of an adversary to generate such set is (much) higher than 2^{64} computations.

8.7. Chosen-Key Distinguishers for 11-round AES-192 and (full) 14-round AES-256

Finally, we present chosen-key distinguishers for 11-round AES-192 and 14-round AES-256. Since the strategies used to set up these distinguishers is similar to the ones proposed for AES-128, we limit ourselves to highlight here the main differences.

8.7.1. Chosen-Key Distinguisher for 11-round AES-192

9-round AES-192 Distinguisher

In order to set up the 9-round distinguisher of AES-192, one exploits the fact that

$$\forall k^4 \in K_{weak}: \qquad \mathcal{IS} \oplus a \xleftarrow{R^{-1}(\cdot)} \mathcal{IS} \xrightarrow{R^2(\cdot)} \mathcal{IS} \oplus b$$

for each key in K_{weak} defined in 4.4.3, where the round constant $R[1]$ that defines K_{weak} must be replaced with $R[4]$.

Chosen-Key Distinguisher for 10-round AES-192

10-round AES-192 Distinguisher - "Weaker" property. As for AES-128, the simplest way to extend the previous distinguisher to 10-round is to exploit a weaker property on (e.g.) the ciphertexts. As a result, *while the property on the plaintexts is unchanged*, the chosen-key model asks the adversary to find a set of 2^{64} (plaintexts, ciphertexts), that is $(p^i, c^i \equiv R^{10}(p^i))$ for $i = 0, ..., 2^{64} - 1$ – where all the plaintexts/ciphertexts are generated by *the same key* – such that *for each* $J \subseteq \{0, 1, 2, 3\}$*, the number of different pairs of ciphertexts that belong to the same coset of* $\mathcal{M}_I$ *is a multiple of* 2.

10-round AES-192 Distinguisher - Freedom of the Key. In order to set up the distinguisher on 10 round, we need a weak invariant subspace trail on 4-round. By simple computation, it is sufficient to choose the subkey[19]

$$\hat{k} \equiv (A = 0, B = 0, C = 0, D = 0, E = 0, F = 0, G = 0, H = 0) \in K_{weak}$$

[18] Note that $\binom{2^{64}+2^{11}}{2^{64}} \geq 2^{32\,823}$.

[19] For completeness, another possible key can be used. In particular, given the key $\hat{k} \in K_{weak}$ defined by $\hat{k} \equiv (A = 0\text{x}63 \oplus R[5], B = 0\text{x}63, C = 0\text{x}63, D = 0\text{x}63, E = 0, F = 0, G = 0, H = 0)$ (where $R[1]$ that defines K_{weak} must be replaced with $R[4]$), then for $\mathcal{IS} \oplus a \xleftarrow{R^{-1}(\cdot)} \mathcal{IS} \xrightarrow{R^3(\cdot)} \mathcal{IS} \oplus b$. We highlight that there is no key that allows to extend both 1-round forward and 1-round backward.

(where $R[1]$ that defines K_{weak} must be replaced with $R[5]$)) for which

$$\mathcal{IS} \oplus a \xleftarrow{R^{-2}(\cdot)} \mathcal{IS} \xrightarrow{R^{2}(\cdot)} \mathcal{IS} \oplus b.$$

Due to the results of Sect. 8.6.1, the multiple-of-128 property (on $\mathcal{D}_J$) and the multiple-of-2 property (on $\mathcal{X}_I$) hold with probability 1 on the plaintexts while the multiple-of-2 property (on $\mathcal{M}_J$) holds on the ciphertexts

$$\textit{Multiple-of-n} \xleftarrow{R^{-3}(\cdot)} \mathcal{IS} \oplus \hat{k} \xleftarrow{R^{-2}(\cdot)} \mathcal{IS} \xrightarrow{R^{2}(\cdot)} \mathcal{IS} \oplus \tilde{k} \xrightarrow{R^{4}(\cdot)} \textit{Multiple-of-}2$$

11-round AES-192 Distinguisher

Finally, it is possible to combine the previous two distinguishers on 10-round AES-192 in order to set up a distinguisher on 11-round AES-192. In this case, the chosen-key model asks the adversary to find a set of 2^{64} (plaintexts, ciphertexts), that is $(p^i, c^i \equiv R^{11}(p^i))$ for $i = 0, ..., 2^{64} - 1$ – where all the plaintexts/ciphertexts are generated by *the same key* – such that the following "*simultaneous multiple-of-n*" property is satisfied:

Plaintext: on the plaintexts, we re-use the previous properties: *(1st)* for each $J \subseteq \{0, 1, 2, 3\}$, the number of different pairs of plaintexts that belong to the same coset of $\mathcal{D}_J$ is a multiple of $128 = 2^7$; *(2nd)* for each $I \subset \{(0, 0), (0, 1), ..., (3, 2), (3, 3)\} \equiv \{(i, j)\}_{0 \le i,j \le 3}$, the number of different pairs of plaintexts that belong to the same coset of $\mathcal{X}_I$ are a multiple of 2;

Ciphertext: *for each* $J \subseteq \{0, 1, 2, 3\}$*, the number of different pairs of ciphertexts that belong to the same coset of* $\mathcal{M}_J$ *is a multiple of* 2.

In order to set up the chosen-key distinguisher, the idea is to exploit the fact that for the key $\hat{k} \in K_{weak}$ defined by $\hat{k} \equiv (A = 0, B = 0, C = 0, D = 0, E = 0, F = 0, G = 0, H = 0) \in K_{weak}$ (where $R[1]$ that defines K_{weak} must be replaced with $R[5]$)), it holds that

$$\mathcal{IS} \oplus a \xleftarrow{R^{-2}(\cdot)} \mathcal{IS} \xrightarrow{R^{2}(\cdot)} \mathcal{IS} \oplus b.$$

Due to the results of Sect. 8.6.1, the multiple-of-128 property (on $\mathcal{D}_J$) and the multiple-of-2 property (on $\mathcal{X}_I$) hold with probability 1 on the plaintexts while the multiple-of-2 property (on $\mathcal{M}_J$) holds on the ciphertexts

$$\textit{Multiple-of-n} \xleftarrow{R^{-3}(\cdot)} \mathcal{IS} \oplus \hat{k} \xleftarrow{R^{-1}(\cdot)} \mathcal{IS} \xrightarrow{R^{2}(\cdot)} \mathcal{IS} \oplus \tilde{k} \xrightarrow{R^{4}(\cdot)} \textit{Multiple-of-}2$$

as required.

What about an adversary facing a family of random and independent ideal ciphers? As we showed in detail in 8.6.4, the required properties on the plaintexts and on the ciphertexts hold with prob. $2^{-32\,823} \simeq 2^{-2^{15}}$ for a random set of texts. Due to our argumentations from 8.6.3, we conjecture that the computational cost of an adversary to generate such set is (much) higher than 2^{64} computations.

8.7.2. Chosen-Key Distinguisher for (*full*) AES-256

Chosen-Key Distinguisher for 12-round AES-256

Similarly, to set up the 12-round distinguisher of AES-256, one exploits the fact that

$$\forall k^4 \in K_{weak} : \qquad \mathcal{IS} \oplus a \xleftarrow{R^{-1}(\cdot)} \mathcal{IS} \xrightarrow{R^{5}(\cdot)} \mathcal{IS} \oplus b$$

for each key in K_{weak} defined in Sect. 4.4.3 where

$$A^0 = A^1 = B^0 = ... = D^0 = D^1 = 0, \quad E^0 = E^1, F^0 = F^1, ..., H^0 = H^1.$$

Chosen-Key Distinguisher for 13-round AES-256

13-round AES-256 Distinguisher - "Weaker" Property. As for AES-128, the simplest way to extend the previous distinguisher to 13-round is to exploit a weaker property on (e.g.) the ciphertexts. As a result, *while the property on the plaintexts is unchanged*, the chosen-key model asks the adversary to find a set of 2^{64} (plaintexts, ciphertexts), that is $(p^i, c^i \equiv R^{13}(p^i))$ for $i = 0, ..., 2^{64} - 1$ – where all the plaintexts/ciphertexts are generated by *the same key* – such that *for each* $J \subseteq \{0, 1, 2, 3\}$*, the number of different pairs of ciphertexts that belong to the same coset of* $\mathcal{M}_I$ *is a multiple of* 2.

13-round AES-256 Distinguisher - Freedom of the Key. Another possibility to extend the previous distinguisher to 13-round is to exploit the freedom in the key. In more details, in order to set up the distinguisher on 13 round and using the same argumentation proposed for AES-128, *among the previous weak-keys* the idea is to choose the sub-key defined by

$$\hat{k} \equiv (E^0 = E^1 = 0\text{x}63 \oplus R[5], F^0 = F^1 = 0\text{x}63, ..., H^0 = 0, H^1 = 0\text{x}63) \in K_{weak}$$

for which

$$\mathcal{IS} \oplus a \xleftarrow{R^{-2}(\cdot)} \mathcal{IS} \xrightarrow{R^5(\cdot)} \mathcal{IS} \oplus b.$$

or

$$\hat{k} \equiv (E^0 = E^1 = F^0 = F^1 = ... = H^0 = 0, H^1 = 0) \in K_{weak}$$

for which

$$\mathcal{IS} \oplus a \xleftarrow{R^{-1}(\cdot)} \mathcal{IS} \xrightarrow{R^6(\cdot)} \mathcal{IS} \oplus b.$$

Chosen-Key Distinguisher on *full* AES-256

The previous chosen-key distinguisher covers 13 rounds of AES-256. Here we show that it is possible to consider a weaker property (e.g.) on the plaintexts to cover full AES-256 in the single-key setting. In this case, the chosen-key model asks the adversary to find a set of 2^{64} (plaintexts, ciphertexts), that is $(p^i, c^i \equiv R^{14}(p^i))$ for $i = 0, ..., 2^{64} - 1$ – where all the plaintexts/ciphertexts are generated by *the same key* – such that the following "*simultaneous multiple-of-n*" property is satisfied:

Plaintext: on the plaintexts, we re-use the previous properties: *(1st)* for each $J \subseteq \{0, 1, 2, 3\}$, the number of different pairs of plaintexts that belong to the same coset of $\mathcal{D}_J$ is a multiple of $128 = 2^7$; *(2nd)* for each $I \subset \{(0, 0), (0, 1), ..., (3, 2), (3, 3)\} \equiv \{(i, j)\}_{0 \le i,j \le 3}$, the number of different pairs of plaintexts that belong to the same coset of $\mathcal{X}_I$ are a multiple of 2;

Ciphertext: *for each* $J \subseteq \{0, 1, 2, 3\}$*, the number of different pairs of ciphertexts that belong to the same coset of* $\mathcal{M}_J$ *is a multiple of* 2.

Choosing the key as before and due to the same arguments given for AES-128 and AES-192, the computational cost to construct such set is of 2^{64}.

What about an adversary facing a family of random and independent ideal ciphers? Due to previous analysis, the required properties holds with prob. $2^{-32\,823} \simeq 2^{-2^{15}}$ for a random set of texts. As before, a simple brute force attack requires at least $2^{64} + 2^{11}$ plaintexts/ciphertexts in order to find a set of 2^{64} plaintexts with the required properties. Due to our argumentations from 8.6.3, we conjecture that the computational cost of an adversary to generate such set is (much) higher than 2^{64} computations.

9

Open Problems - Cryptanalysis of AES

Subspace Trails

- In Sect. 4, we propose the definition of "weak-key subspace trail". A natural question arises: is it possible to set up a proper weak-key subspace trail - that is, different both from an invariant subspace trail (i.e. with disjoint input and output subspaces) and different from a subspace trail (i.e. that works only for a class of weak keys) - that improves/outperforms the ones in such section? A possible starting point could be to the results given in [BWP05] about the several subspaces $V, W \subset GF(2^8)$ of dimension two or/and four that satisfy $V \neq W$ and $\text{Sbox}(V \oplus v) \subseteq W \oplus w$ (where $\text{Sbox}(x) = x^{-1}$).

Truncated Differential Distinguishers/Attacks on 5-round AES

- In order to theoretically describe the 5-round AES truncated differential distinguisher based on the mean proposed in Sect. 5, we need particular assumptions on the S-Box. As we shown in details there, such assumption is related to the fact that the number of solutions x of the equation
$$\text{S-Box}(x \oplus \Delta_I) \oplus \text{S-Box}(x) = \Delta_O$$
are uniformly distributed for each input/output difference $\Delta_I \neq 0$ and/or $\Delta_O \neq 0$. In there we observed the following: given the probabilistic distribution of the number of solutions of such equation for each $\Delta_I \neq 0$ and/or $\Delta_O \neq 0$, its variance is "low" if the solutions are uniform distributed.

  On the other hands, consider the practical results on small-scale AES proposed in Sect. 5.8. The just given property on the variance of such distribution does *not* seem sufficient to guarantee that the solutions are uniform distributed. To better understand this fact, consider the average number of collisions for different "S-Boxes with the same variance". By Table 5.2, it turns out that the average number of collisions is not (approximately) equal for all the "S-Boxes with the same variance", as we should expect.

  An open problem is to understand *which parameters/properties of the S-Box really influences the fact that the solutions of the equation (5.5) are uniformly distributed for each input/output difference. Is it possible to better estimate the average number of collisions taking into account these parameters/properties of the S-Box?* In other words, is it possible to theoretically compute a confidence interval $[M - m, M + m]$ which depends on the details of the S-Box and such that all the number of collisions given in Table 5.2 - Sect. 5.8 fall into it (with high probability)?

- The 5-round AES truncated differential distinguisher based on the mean seems to depend also on the details of the MixColumns matrix. It could be interesting to better understand this fact: *which details of the MixColumns matrix* – especially in the case of "bad" S-Box – *influence the average number of collisions for 5-round AES? Is it possible to formally compute the mean – or equivalently, re-formulate the proof proposed in Sect. 5.3 – using a "weaker" assumption* *than the MDS one? Is it possible to theoretically predict the average number of collisions when working with a matrix*[1] *which is not MDS?*

- In Sect. 5.7.1, we propose the first truncated differential distinguisher based on the variance which is (much) more competitive - both for the computational and data costs - than the corresponding one based on the mean. However, an open problem is *to formally study* the probability of success of such distinguisher. Is it possible to set up similar distinguishers for other ciphers?

- As highlighted in Sect. 5.6.2 , the skewness of the probabilistic distribution of 5-round AES is different from the one of a random permutation. In particular, the bias in the skewness seems to be stronger than the bias in the mean. As a result, also this parameter could be potentially used to set up distinguishers and/or key-recovery attacks. The problem *to theoretically compute the value of the skewness* of such probabilistic distributions is open for future research: does it depend on the details of the S-Box? Is it possible to exploit such parameter also for other ciphers? What about the kurtosis?

- Let's focus again on the 5-round AES truncated differential distinguisher based on the mean, in particular on its proof. In order to theoretically compute the average number of collisions, we focus and work only on the middle round of

  $$\mathcal{D}_I \oplus a \xrightarrow[\text{prob. } 1]{R^2(\cdot)} \mathcal{M}_I \oplus b \xrightarrow{R(\cdot)} \mathcal{D}_J \oplus a' \xrightarrow[\text{prob. } 1]{R^2(\cdot)} \mathcal{M}_J \oplus b'.$$

  In particular, given two plaintexts $p^1, p^2 \in \mathcal{M}_I \oplus b$, the fact that they belong to the same coset of $\mathcal{D}_J$ after one round - that is, $R(p^1) \oplus R(p^2) \in \mathcal{D}_J$ - can be re-written as a system of equations that involves the S-Box operation of the form

  $$\bigoplus_{\{i,j\} \in \mathcal{I}} A_{i,j} \times \left[\text{S-Box}(B_{i,j} \cdot p^1_{i,j} \oplus C_{i,j}) \oplus \text{S-Box}(B_{i,j} \cdot p^2_{i,j} \oplus C_{i,j})\right] = 0$$

  for a set of index $\mathcal{I}$ and for some constants A, B, C (which depend only on the MixColumns matrix and on the secret key).

  A similar analysis can be performed for 6-round AES, by replacing the S-Box operation with the *super-Sbox* one (3.1). Indeed, note that *(1st)* 6-round AES can be re-written as

  $$\mathcal{D}_I \oplus a \xrightarrow[\text{prob. } 1]{R^2(\cdot)} \mathcal{M}_I \oplus b \xrightarrow{R^2(\cdot)} \mathcal{D}_J \oplus a' \xrightarrow[\text{prob. } 1]{R^2(\cdot)} \mathcal{M}_J \oplus b',$$

  and *(2nd)* given two plaintexts $p^1, p^2 \in \mathcal{M}_I \oplus b$, the fact that they belong to the same coset of $\mathcal{D}_J$ after two rounds - that is, $R^2(p^1) \oplus R^2(p^2) \in \mathcal{D}_J$ - can be re-written as a system of equations that involves the *super-Sbox* operation (3.1), similar to the one just given. An open problem is to modify the proof given in Sect. 5.3 in order to get a similar result about the average number of collisions for 6-round AES. Is it possible to set up a truncated differential distinguisher for 6-round AES which is independent of the secret key?

- In Sect. 5.9, we presented several key-recovery attacks on 5-round AES based on 4-round truncated differential distinguishers (which are independent of the secret key). Is it possible to improve the computational and/or data costs of such attacks? Is it possible to extend such attacks to 6-round AES without guessing an entire subkey? Is it possible to set up similar attacks on 6- (or even more) round AES-128 exploiting *directly* the 5-round truncated differential distinguishers based on the mean and on the variance?

[1] Note that the MixColumns matrices of many AES-like lightweight ciphers in the literature are not MDS.

- Since reduced versions of AES have nice and well-studied properties, many constructions employ round-reduced AES as part of their design. E.g., several candidates in the on-going "Competition for Authenticated Encryption: Security, Applicability, and Robustness" (CAESAR) are designed based on an AES-like SPN structure. As an open future problem, it could be interesting to apply the distinguishers and key-recovery attacks presented before to tweakable block-ciphers based on AES. Is it possible to extend them for more rounds than the ones previous given for AES, using the freedom of the tweak or exploiting related-tweak attacks?

Mixture Differential Cryptanalysis

- In Sect. 6, we proposed several probabilistic mixture differential distinguishers on 5-round AES which are independent of the secret key. All these distinguishers have been obtained by combining the 4-round mixture differential distinguisher with a truncated differential trail at the end. Is it possible to improve the computational and/or data costs of such distinguishers? Is it possible to set up (competitive) key-recovery attacks based on them? Is it possible to set up similar distinguishers for other ciphers? Is it possible to use the same strategy to set up a distinguisher on 6-round by combining the 4-round mixture differential distinguisher with an impossible (truncated) differential trail?

- Mixture differential cryptanalysis proposed in Sect. 6 is a new technique that allows to set up competitive distinguishers and key-recovery attacks on round-reduced AES. Since it has been proposed only recently, a deep analysis of its potential is still missing. As a result, several problems/questions are open for future research: Is it possible to exploit it to attack other construction, like AES-PRF proposed in [MN17]? Is it possible to combine it with a boomerang/yoyo distinguisher/attack?

AES with Secret S-Box

- Is it possible to combine Mixture Differential Cryptanalysis and the strategy presented in [TKKL15] (based on the integral cryptanalysis) in order to set up competitive key-recovery attacks on AES with a single secret S-Box?

- In Sect. 7, we presented and proposed several attacks on AES with a single secret S-Box. Is it possible to extend them on more rounds of AES? What about the case in which AES is instantiated by different secret S-Boxes? Is it possible to set up similar attacks in the case of AES with known S-Box and secret linear layer?

Open-Key Distinguisher on AES

- As highlighted in Sect. 8, an open problem for the open-key distinguisher is to find a (formal) definition for known-/chosen-key distinguishers on concrete implementable cipher (like the AES), that captures the idea of the distinguishers present in the literature. A possible way to achieve this result is to find a formal definition of the set $\mathfrak{D}$ of distinguishers as proposed in Sect. 8.1.2

  $\mathfrak{D}$ denotes the set of all distinguishers D for which the description of the generic relation $\mathcal{R}$ has no "obvious connection" with the specification of E and it is independent of the value of the key.

- Is it possible to set up a known-key distinguisher on full AES-128 which does not exploit Gilbert's model/scenario? Is it possible to set up other (more competitive) chosen-key distinguishers on (full) AES and/or to provide a formal proof that support the distinguishers presented here?

- Interestingly, while it is possible to set up chosen-key distinguishers on full AES-128 and on full AES-256, this is not possible for AES-192. A possible reason of this is the the key-schedule.

  In some sense, the key-schedule of AES-192 seems to be "stronger" than the one of AES-256. The interesting fact is that, even if one has more freedom for AES-192 than AES-128, it is not possible to set up distinguishers longer for AES-192 than for AES-128. To explain this fact, consider the key-schedule of AES-192 in detail, and let's focus on the invariant subspace trail of AES proposed in Sect. 4. Even if one has 64-bit of freedom more than AES-128, these bits are not "free", in the sense that they are forced to take particular values in order to fulfill the invariant subspace trail. In the case of AES-256 instead, one can completely exploit the 128-bit of freedom in order to extend the distinguisher for 3 more rounds.

  As open problem, it could be interesting to consider possible variant of the key-schedule of AES-256 that guarantees the same security offered by the one of AES-192. A possible way to achieve this result could be to break the symmetry of the key-schedule of AES-256.

Part II.

Novel Designs: MiMC and its Generalizations

10
MiMC

Modern cryptography developed many techniques that go well beyond solving traditional confidentiality and authenticity problems in two-party communication. Secure multi-party computation (MPC), zero-knowledge proofs (ZK), and fully homomorphic encryption (FHE) are only some of the most striking examples. In various applications of these three technologies, part of the circuit or function that is being evaluated is in turn a cryptographic primitive such as a PRF, a symmetric encryption scheme, or a collision resistant function. As a result, a new direction of research aims at designing symmetric schemes which are competitive for these particular applications.

Traditionally, ciphers are built from linear and non-linear building blocks. These two have roughly similar costs in hardware and software implementations. In CMOS hardware, the smallest linear gate (XOR) is about 2-3 times larger than the smallest non-linear gate (typically, NAND). When implemented in an MPC protocol or a homomorphic encryption scheme, however, the situation is radically different: linear operations come almost for free, whereas the bottleneck are nonlinear operations that involve symmetric cryptographic operations.

This cost metric suggests *a new way of designing a cipher* where most of the cryptographically relevant work would be performed as linear operations and the use of non-linear operations is minimized. This design philosophy is related to the fundamental theoretical question of the minimal multiplicative complexity (MC) [BPP00] of certain tasks.

In [AGR+16; AGP+18], we focus on a large class of such applications where the total number of field multiplications in the underlying cryptographic primitive poses the largest performance bottleneck. Examples include MPC protocols based on Yao's garbled circuit and ZK-proof systems, including developments around SNARKs [BCG+13] which found practical applications, e.g., in Zerocash [BCG+14]. This motivates the following question addressed in this work: *How does a construction for a secure block cipher or a secure cryptographic hash functions look like that minimizes the number of field multiplications?*

MiMC: "Minimize the Multiplicative Complexity"

To answer this question, here we present[1] MiMC [AGR+16], which design is extremely simple: a non-linear function $F(x) := x^3$ is iterated with subkey additions. This design is a simplified variant of a design by Nyberg and Knudsen [NK95] from the 1990s, which was aimed to demonstrate ways to achieve provable security against the emerging differential and linear attacks, using a small number of rounds (smaller than, say, DES). However, not much later, [JK97] showed very efficient, even practical interpolation attacks on such proposals. Our proposal resembles $\mathcal{PURE}$, a design introduced in [JK97] in order to present the interpolation attack. We pick up this work from almost 20 years ago and study if a much higher number of rounds can make (a simplified version of) this design secure. It turns out, perhaps surprisingly, that the required much higher number of rounds (in the order of 100s instead of 10 or less) is very competitive when it comes to the new application areas of symmetric cryptography that motivate this work.

[1] The name MiMC is an abbreviation of "*Mi*nimize the *M*ultiplicative *C*omplexity", which is exactly the goal of this cipher.

MiMC - which can be instantiated both in $GF(p)$ and in $GF(2^n)$ - can be used for encryption as well as for collision-resistant cryptographic hashing based on a sponge construction. MiMC is distinguished from any of the many constructions that have been proposed in this field recently, and it contradicts a popular belief: a recent standard textbook [KR11, Sect. 8.4] explicitly considers such constructions as "*not serious, for various reasons*".

Related Works and Comparison

Recently, a number of new primitives were proposed that aim to minimize metrics related to the computation of multiplications - we refer to Sect. 2.4 for a discussion about this topic. Among others, they include LowMC [ARS+15], Kreyvium [CCF+16], FLIP [MJSC16] and RASTA [DEG+18]. While LowMC has been designed for FHE, MPC or/and ZK applications, in contrast Krevyium, FLIP and RASTA have been designed mainly for FHE application. Since the main targets of MiMC are Zero-Knowledge proofs and MPC applications (rather than FHE ones), here we limit ourselves to recall only LowMC in details.

LowMC [ARS+15]. LowMC is based on a parameterizable design approach. It is a flexible block cipher based on an SPN structure where, given any blocksize, a choice for the number of S-Boxes per round and security expectations in terms of time and data complexity, a concrete instantiation can be created easily (the number of rounds needed to reach the security claims is indeed derived from these parameters).

In more details, to reduce the multiplicative complexity, the number of S-Boxes applied in parallel can be reduced, leaving part of the substitution layer as the identity mapping. Such a strategy was not new in the literature (e.g. it was already proposed in ZORRO [GGNS13]), and despite several concerns regarding it [BDD+15], authors showed that security is viable. To reach security in spite of a low multiplicative complexity, pseudo-randomly generated binary matrices are used in the linear layer to introduce a very high degree of diffusion.

As just recalled, LowMC has been designed for applications like FHE, MPC, or/and ZK. In [ARS+15], authors gave several instatiations in order to target each one of these applications, e.g. some that minimize the ANDdepth, others that minimize the number of ANDs overall, and again others that minimize the number of ANDs per encrypted bit.

MiMC Motivations. Earlier works on specialized designs for such applications - including LowMC, Kreyvium, FLIP or the very recent RASTA - consider the case of Boolean multiplications and mostly focus on the depth of the resulting circuit.

Surprisingly, albeit many MPC/FHE/ZK-protocols natively support operations in $GF(p)$ for large prime p, very few candidates (even considering all of symmetric cryptography) exist which natively work in such fields. For this reason, in [AGR+16] we decided to focus on multiplications in the larger fields $GF(2^m)$ and $GF(p)$ which is motivated as follows: *as many protocols support multiplications in larger fields natively, encoding of a description in $GF(2)$ is cumbersome and inefficient.* Whilst it is possible to do bit operations over $GF(p)$ using standard tricks (which turn XOR into a non-linear operation), such a conversion is expensive. Consider AES as an example: it allows for an efficient description in a variety of field sizes. This is also the reason why the bit-based LowMC, which has a lower number of AND gates, can often outperform AES in actual implementations of the MPC protocols, despite being much better than AES in terms of $GF(2)$ metrics (see [ARS+15, Table 6] for details of the most striking example). This is also partly due to the *very* high number of XORs computed in LowMC resulting them to be no longer negligible.

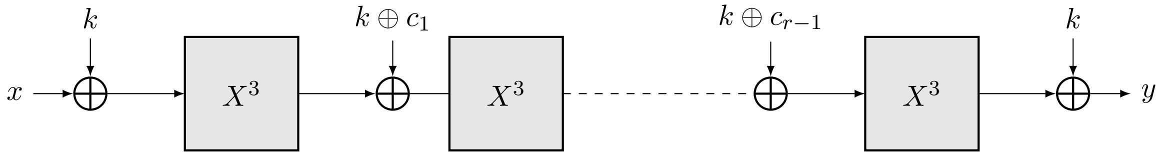


Figure 10.1.: r rounds of MiMC-n/n.[2]

10.1. The MiMC Primitives

In the following, we describe a block cipher, a permutation, and a (sponge-based) hash function with a low number of multiplications in a finite field $\mathbb{F}_q \equiv GF(q)$ where q is either a prime p or a power of 2.

10.1.1. The Block Cipher MiMC

In order to achieve an efficient implementation over a field $\mathbb{F}_q$ (with q either prime or a power of 2), i.e. to minimize computationally expensive multiplications in the field, our design operates entirely over $\mathbb{F}_q$, thereby avoiding S-Boxes completely. More precisely, we use a permutation polynomial over $\mathbb{F}_q$ as round function.

In the following, we restrict ourselves to $\mathbb{F}_{2^n}$ and we denote by MiMC-N/κ a keyed permutation with block size N and key size κ. The concept however equally applies to $\mathbb{F}_p$.

MiMC-n/n

Our block cipher is constructed by iterating a round function r times where each round consists of three steps:

- a key addition with the key k;
- the addition of a round constant $c_i \in \mathbb{F}_{2^n}$;
- the application of a non-linear function defined as $F(x) := x^3$ for $x \in \mathbb{F}_{2^n}$.

The ciphertext is finally produced by adding the key k again to the output of the last round. Hence, the round function is described as

$$F_i(x) = F(x \oplus k \oplus c_i)$$

where $c_0 = c_r = 0$ and the encryption process is defined as

$$E_k(x) = (F_{r-1} \circ F_{r-2} \circ ... \circ F_0)(x) \oplus k.$$

In order to guarantee invertibility, we choose n to be odd. As we are going to show, the number of rounds to provide security is given by

$$r = \lceil n \cdot \log_3 2 \rceil + 1,$$

where the $r - 1$ round constants are chosen as random elements from $\mathbb{F}_{2^n}$.

Note that the random constants c_i do not need to be generated for every evaluation of MiMC. Instead the constants are fixed once and can be hard-coded into the implementation on either side. No extra communication is thus needed, just as with round constants in LowMC, AES, or in fact any other cipher.

[2] ***Acknowledgement.*** *Figure 10.1 – made by Christian Rechberger and by Arnab Roy – has been copied from [AGR+16].*

Decryption for MiMC-n/n can be realized analogously to encryption by reversing the order of the round constants and using

$$F^{-1}(x) := x^s \quad \text{where} \quad s = \frac{2^{n+1}-1}{3}$$

instead of $F(x) := x^3$. Hence, encryption and decryption need to be implemented separately. Furthermore, since decryption is much more expensive than encryption, using modes where the inverse is not needed is advisable. We note that for our targeted applications, such as PRFs or cryptographic hash functions, computing the inverse is usually not required. We therefore provide benchmark results only for the encryption function. The fact that the inverse has a more complex algebraic description also has a beneficial effect on security as it limits cryptanalytic approaches that try to combine the encryption and decryption direction, such as inside-out approaches.

Design Rationale. Here, we briefly we explain the design rationale of the keyed permutation and argue its security. The monomial x^3 serves as the non-linear layer of the block cipher. Note that we can use x^3 to construct the cipher iff it is a permutation monomial in the field $\mathbb{F}_{2^n}$. The following well known result governs the choice of the monomial and size of the field in the design of MiMC.

Proposition 24. *Any monomial x^d is a permutation in the field $\mathbb{F}_{2^n}$ iff* $\gcd(d, 2^n-1) = 1$.

Hence, $x \to x^3$ is a permutation in $\mathbb{F}_{2^n}$ only when n is odd.

In order to compute the inverse of x^3 in $\mathbb{F}_{2^n}$, the goal is to find an exponent s s.t. $x^{3\cdot s} = x$. By Fermat's Little Theorem[3], this is equivalent to look for an s such that $3 \cdot s = 1 \pmod{2^n - 1}$. By previous proposition, remember that 3 divides $2^x - 1$ if and only if x is even. As a result, we have that

$$s = \frac{2 \cdot (2^n-1)+1}{3} = \frac{2^{n+1}-1}{3}.$$

Lemma 9. *Let n be an odd number. The inverse of the cubic function $f(x) : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ defined as $f(x) = x^3$ is given by*

$$f^{-1}(x) = x^{\frac{1}{3}} \equiv x^{\frac{2^{n+1}-1}{3}}.$$

In a similar way (see "Hermite's Criterion" for more details):

Proposition 25. *The function $f(x) = x^3$ is a permutation in the field $\mathbb{F}_p$ iff $p \neq 1 \mod 3$.*

10.1.2. The Hash Function – MiMCHash

First of all, note that it is possible to construct a permutation MiMC^P from the cipher MiMC as described above by simply set the key to e.g. the all-0 string.

For the hash function MiMCHash, we propose to instantiate the permutation MiMC^P in the sponge framework [BDPA07]. When the internal permutation P of an N-bit sponge function (composed of c-bit capacity and r-bit bitrate - $N = c + r$) is modeled as a randomly chosen permutation, it has been proven by Bertoni *et al.* [BDPA08] to be indifferentiable from a random oracle up to $2^{c/2}$ calls to P. In other words, a sponge with a capacity of c provides $2^{c/2}$ collision and $2^{c/2}$ (second) preimage resistance. Given a permutation of size n, and a desired security level s, we can hash $r = n - 2s$ bits per call to the permutation. MiMCHash-l denotes the hash function with l bit output.

As usual, the message is first padded according to the sponge specification so that the number of message blocks is a multiple of r where r is the rate in sponge mode. For MiMCHash-t we use MiMC-n/n permutation where $n = 4 \cdot t + 1$ and $s = 2 \cdot t$. For MiMCHash-256 we thus use a MiMC-n/n permutation with $n = 1025$. The rate and the capacity are chosen as 512 and 513 respectively. This choice allows for processing the same amount of input bits as SHA-256 (512 bits) while at the same time offering collision security and (second) preimage security of 256 bits.

[3] Fermat's little theorem states that if p is a prime number, then for any integer a, the number $a^p - a$ is an integer multiple of p. In the notation of modular arithmetic, this is expressed as $a^p \equiv a \mod p$.

10.2. Security Analysis

Our designs resist a variety of cryptanalysis techniques. The algebraic design principle of MiMC causes a natural concern about the security of the keyed permutation against algebraic cryptanalytic techniques. We describe several possible algebraic attacks (including a new "GCD" attack) against the design and analyze the resistance of the block cipher against these attacks. We also consider statistical attacks.

To summarize the following results, the number of rounds for MiMC-n/n is derived from an interpolation attack.

10.2.1. Interpolation Attack

Interpolation attacks, introduced by Jakobsen and Knudsen [JK97], construct a polynomial corresponding to the encryption func- tion without knowledge of the secret key. If an adversary can construct such a polynomial then for any given plaintext the corresponding cipher-text can be produced without knowledge of the secret key.

Let $E_k : \mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n}$ be an encryption function. For a randomly fixed key k, the polynomial $P(x)$ representing $E_k(x)$ can be constructed using Lagrange's theorem, where x is the indeterminate corresponding to the plaintext. If the polynomial has degree d then we can find it using Lagrange's formula

$$P(x) = \prod_{i=0}^{d} y_i \sum_{1 \leq j \leq d, i \neq j} \frac{x - x_j}{x_i - x_j}$$

where $E_k(x_i) = y_i$ for $i = 0, 1, ..., d$.

This method can be extended to a key recover attack. The attack proceeds by simply guessing the key of the final round, decrypting the cipher-texts and constructing the polynomial for $r - 1$ rounds. With one extra p/c pair, the attacker checks whether the polynomial is correct.

Observe that the number of unknown coefficients of the interpolation polynomial is $d + 1$ and that the complexity of constructing a Lagrangian interpolation polynomial is $\mathcal{O}(d \log d)$ [Sto85]. Hence, setting $d = 3^r$ with

$$r = r_{max} \approx n / \log_2(3)$$

thwarts this attack. Note that no function mapping from $GF(2^n)$ to $GF(2^n)$ has degree $\geq 2^n$, since $T^{2^n - 1} \equiv 1$ for each $T \in \mathbb{F}_{2^n}$ and the degree of the interpolation polynomial does not increase for $r > r_{max}$.

A meet-in-the-middle variant of the interpolation attack was also proposed in [JK97], constructing a polynomials $g(x) = h(y)$ instead of one polynomial $y = f(x)$. However, for MiMC-n/n, this approach does not produce an improvement due to the prohibitive degree of the inverse operation.

For completeness, we note that the complexity of an interpolation attack may decrease if the polynomial $P(x)$ is sparse for a chosen key. However, because we are adding random round constants in each round and x^3 is a permutation in $\mathbb{F}_{2^n}$ by construction, our $P(x)$ is not expected to be sparse[4].

[4] This claim is supported by our experiments. In particular, for a field $\mathbb{F}_{2^n}$ and using x^3 as permutation, we observed:

- after 1 round, all terms appear (percentage: 100 %);
- after 2 rounds, 8 terms appear instead of 10 (percentage: 80 %);
- after 3 rounds, 19 terms appear instead of 28 (percentage: 67.86 %);
- after 4 rounds, 54 terms appear instead of 82 (percentage: 65.85 %);
- after 5 rounds, 161 terms appear instead of 244 (percentage: 66 %);
- after 6 rounds, 531 terms appear instead of 730 (percentage: 72.74 %);

and so on, where the percentage of the non-null terms continues to grow for the next rounds. For example, for the concrete field $GF(2^{17})$, after 10 rounds almost all the terms are non-zero.

10.2.2. GCD Attack

From the description of MiMC, it is clear that factoring univariate polynomials recovers the key. However, if we are given more than one known plaintext-cipher-text pair, we can reduce the complexity further by computing a GCD of them.

Denote by $E(k, x)$ the encryption of x under key k. For a pair $(x, y) \in \mathbb{F}_q \times \mathbb{F}_q$, $E(K, x) - y$ denotes a univariate polynomial in $\mathbb{F}_q[K]$ corresponding to (x, y). Note that in general, given plaintext/cipher text pair (x, y), it should be hard for a generic encryption scheme to compute the univariate polynomial $E(K, x) - y$ explicitly in the variable K (i.e. the secret key). However, this is not the case of MiMC, for which the polynomial $E(K, x) - y$ can be always computed explicitly, and it simply corresponds to the definition of encryption process (that is, the iterative application of the cubic function). Moreover, note that this attack may also be applied to $\mathcal{PURE}$, the cipher used in [JK97] to demonstrate the vulnerability of the KN cipher to interpolation attacks, assuming round keys are not independent but linearly derived from k.

Consider now two such polynomials $E(K, x_1) - y_1$ and $E(K, x_2) - y_2$, with $y_1 = E(k, x_1)$ and $y_2 = E(k, x_2)$ for the fixed but unknown key k. It is clear that these polynomials share $(K - k)$ as a factor. Indeed, with high probability the greatest common divisor will be $(K - k)$. Thus, by computing the GCD of the two polynomials, we can find the value of k.

MiMC-n/n for a known plain text x corresponds to a polynomial having degree 3^r, where the leading monomial always has non-zero coefficient. Hence, we can recover k with a GCD computation of two polynomials at degree 3^r (indeed, considering differences of two polynomials $G(K, x_i) - y_i$ reduces this degree to $3^r - 1$ by canceling the leading term). It is well-known that the complexity for finding the GCD of two polynomials of degree d is $\mathcal{O}(d \log^2 d)$. Hence, the complexity of this attack is $\mathcal{O}(r^2 \cdot 3^r)$. As a result, for MiMC-$n/n$ the time complexity of this attack is higher than that of the interpolation attack.

10.2.3. Algebraic Degree and Higher-Order Differentials

A well-known result from the theory of Boolean functions is that if the algebraic degree of a vectorial Boolean function $f(\cdot)$ (like a permutation) is d, then the sum over the outputs of the function applied to all elements of a vector space $\mathcal{V}$ of dimension $\geq d + 1$ is zero (as is the sum of all inputs, i.e., the elements of the vector space). The same property holds for affine vector spaces of the form $\{v + c \,|\, v \in \mathcal{V}\}$ for arbitrary constant c

$$\bigoplus_{v \in \mathcal{V} \oplus c} v = \bigoplus_{v \in \mathcal{V} \oplus c} f(v) = 0.$$

This is the property exploited by higher-order differential attack [Knu94].

As discussed above, the large number of rounds ensures that the algebraic degree of MiMC in its native field will be maximal or almost maximal. This naturally thwarts higher-order differential attacks [Knu94] when considering the difference as defined in the field (i.e., using the inverse of the field addition).

What happens to the degree when viewing the rounds as vectorial Boolean functions? As squaring is a linear operation in $\mathbb{F}_{2^n}$, it is also linear when viewed as vectorial function over $\mathbb{F}_2$. Cubing on the other hand introduces an additional multiplication which gives the round function an algebraic degree of 2 in every component when viewed as a vectorial Boolean function. Thus, the large number of rounds should cause the degree to rise quickly and reach the limit of 2^n which is sufficient to thwart any higher-order differential attacks also when viewing the round function as a vectorial Boolean function.

10.2.4. Statistical and Other Attacks

Differential Attacks

Differential cryptanalysis [BS90; BS93] is one of the most widely used technique in symmetric-key cryptanalysis. The different types of cryptanalysis methods based on this technique depend on the propagation of an input difference through a given number of rounds of an iterative block cipher to yield a known output difference with high probability. The probability of the propagation often determines how many rounds can be attacked using this technique.

Given an input difference δ_I and an output difference δ_O, the differential probability of the round function is given as

$$Prob(\delta_I \to \delta_O) = \frac{|\{x \in \mathbb{F}_{2^n} \mid F(x \oplus \delta_I) \oplus F(x) = \delta_O\}|}{2^n}.$$

In our case the number of x satisfying $F(x \oplus \delta_I) \oplus F(x) = \delta_O$ is determined by the non-linear function x^3. Hence it is enough to determine the size of the set

$$D = \{x \in \mathbb{F}_{2^n} \mid (x \oplus \delta_I)^3 \oplus x^3 = \delta_O,\, \delta_O \neq 0\}$$

As this is a quadratic equation in x for any δ_I and $\delta_O \neq 0$, there are at most two solutions to the equation. This implies $Prob(\delta_I \to \delta_O) \leq 2^{-n+1}$. It follows that this is sufficient to give any differential trail of at least two rounds a probability too low to be useful in an attack. A detailed analysis of the differential property of monomials of the form x^{2^t+1} in $\mathbb{F}_{2^n}$ can be found e.g. in [Nyb94].

Linear Attacks

Similar to differential attacks, linear attacks [Mat93] pose no threat to MiMC. Indeed, the cubic function is an almost bent or an almost perfect nonlinear (APN) function, i.e., differential 2-uniform, where an APN permutation provides the best resistance against linear and differential cryptanalysis. Thus, since its maximum square correlation is limited to 2^{-n+1} (cf. for example [AÅBL12] for details), any linear trail of the cubing function will have negligible potential after a few rounds.

Invariant Subfields

The algebraic structure of MiMC allows to mount a invariant subfield attack on the block cipher under a poor choice of round constants. That is, if all the round constants c_i and the key k are in subfield $\mathbb{F}_{2^m}$ of $\mathbb{F}_{2^n}$ then by choosing a plaintext $x \in \mathbb{F}_{2^m}$ an adversary can ensure that $E_k(x) \in \mathbb{F}_{2^m}$. This attack is thwarted by picking n to be prime. The only subfield is then $\mathbb{F}_2$ such that picking constants $\neq 1$ will be enough to avoid the attack.

10.2.5. Hash-Specific Security Considerations

For usage in the MiMC permutation in the sponge mode as described in Sect. 10.1.2 we require the permutation to not show non-trivial non-random behavior for up to 2^s input/output pairs. As specified in Sect. 10.1.2 the size of the permutation n determines the number of rounds (based on the GCD attack described above). As $2s < n$, this choices leaves us with an additional security margin.

In more details, given a sponge construction instantiated by the MiMC permutation, the number of rounds of the inner permutation is chosen according to number of rounds of MiMC. This is due to the following considerations. First, as we just recalled, when the internal permutation $\mathcal{P}$ of an $N = c + r$ bit sponge function is modeled as a randomly chosen permutation, the sponge hash function is indifferentiable from a random oracle up to $2^{c/2}$ calls to $\mathcal{P}$. The numbers of rounds of MiMC has been chosen in order to guarantee security against any (secret-/known-/chosen-) distinguisher which is independent of the key. Equivalently, this means that such number of rounds guarantee that $\mathcal{P}$ does not present any non-random/structural property (among the ones known in the literature[5]). It follows that the previous assumption is satisfied. These and the fact that every key-recovery attack is meaningless in the hash scenario support our choice to consider the univariate case in order to determine the number of rounds of the inner permutation. For completeness, we remark that the fact that $\mathcal{P}$ presents a non-random/structural property does not imply an attack on the hash sponge function instantiated by $\mathcal{P}$.

10.3. Variants

10.3.1. MiMC over Prime Fields

MiMC can also be used to operate over prime fields i.e. a field $\mathbb{F}_p$ where p is prime. First of all, in that case, it needs to be assured that the cubing in the round function creates a permutation. For this, it is sufficient to require $\gcd(3, p-1) = 1$.

Following the notation as above, we can consider MiMC-p/p where the permutation monomial x^3 is defined over $\mathbb{F}_p$. The number of rounds for constructing the keyed permutation is $r = \log_3(\log_2 p)$ (note that $p \approx 2^n$), and the r round constants are chosen as random elements in $\mathbb{F}_p$.

Our cryptanalysis just proposed transfers to this case, except for the subfield attack which does not apply here.

10.3.2. Different Round Functions

Considering the case $GF(2^n)$, one may consider a round function of the form

$$F(x) = (x \oplus k \oplus c)^d$$

for generic exponents d. Our choice to consider the exponent 3 is due to the following analysis.

For simplicity, we limit ourselves to consider exponents of the form $2^t + 1$ and $2^t - 1$, for positive integer t (note that 3 is the only number that can be written in both ways), which are for different reasons - as explained in the following - the best candidates. Remember that for MiMC-n/n, d has to satisfy the condition $\gcd(d, 2^n - 1) = 1$ in order to be a permutation.

For further analysis, we recall the Lucas's Theorem.

Theorem 11 (Lucas's Theorem). *For non-negative integers m and n and a prime p, the following congruence relation holds:*

$$\binom{m}{n} = \prod_{i=0}^{k} \binom{m_i}{n_i} \mod p$$

where $m = m_k \cdot p^k + m_{k-1} \cdot p^{k-1} + ... + m_1 \cdot p + m_0$ and $n = n_k \cdot p^k + n_{k-1} \cdot p^{k-1} + ... + n_1 \cdot p + n_0$ are the base p expansions of m and n respectively, using the convention that $\binom{x}{y} = 0$ if $x < y$.

Exponents of the form $\mathbf{2^t + 1}$. Exponents of the form $2^t + 1$ (with $t > 1$) have the nice property that the cost to compute x^{2^t+1} does not depend on t since any square operation is “linear” – in the sense that it satisfies the property $(a + b)^2 = a^2 + b^2$ – in $GF(2^n)$, that is it requires only one multiplication independently of t. Moreover, the degree of the resulting r-round interpolation polynomial is $(2^t + 1)^r$, which is significantly higher than 3^r even for “small” t. On the other hand, the major problem of these kind of exponents is that the corresponding interpolation polynomials

[5] That is, we do not exclude that a non-random property can be discovered in the future.

are in general sparse. E.g. using Lucas's Theorem, note that just after one round the interpolation polynomial has only 4 terms instead of $2^t + 2$:

$$(x \oplus k)^{2^t+1} \equiv_2 (x \oplus k)^{2^t} \cdot (x \oplus k) \equiv_2 (x^{2^t} \oplus k^{2^t}) \cdot (x \oplus k) \equiv_2 x^{2^t+1} \oplus x^{2^t} \cdot k \oplus k^{2^t} \cdot x \oplus k^{2^t+1}$$

Using the same technique, after r rounds, the number of terms of the polynomial is upper bounded by $3^r + 1$, which is (much) smaller than $(2^t + 1)^r + 1$. In particular, note that $3^r + 1$ is exact the same upper bounded obtained for the exponent 3 (which corresponds to $t = 1$). As a result, the number of rounds to guarantee the security against the algebraic attacks does not change choosing exponent of the form $2^t + 1$ for $t > 1$ - remember that the number of plaintexts/ciphertexts required in order to construct the interpolation polynomial depends on the number of terms of that polynomial[6]. That is, both from the security point of view and from the implementation one, there is no advantage to choose exponents of the form $2^t + 1$ greater than 3. Similar considerations can be done also for exponents of the form $2^t + 2^s = 2^s \cdot (2^{t-s} + 1)$, where $s < t$.

Exponents of the form $\mathbf{2^t - 1}$. Consider now the case of exponents of the form $2^t - 1$. Such exponents provide security against interpolation and algebraic attacks in general, since the corresponding interpolation polynomials are not sparse:

$$(x \oplus k)^{2^t-1} \equiv_2 \bigoplus_{i=0}^{2^t-1} x^i \cdot k^{2^t-1-i}$$

since

$$\forall i = 0, ..., 2^t - 1 : \qquad \binom{2^t - 1}{i} \equiv_2 1.$$

One the other hand, one has to compute more multiplications and square operations in order to calculate x^{2^t-1} w.r.t. x^{2^t+1}. Thus, under the assumption that x^2 is linear, one may ask the following: *is it possible to minimize the total number of multiplications necessary to compute the ciphertext choosing an exponent of the form $2^t - 1$ different from 3?*

There are different ways to compute g^e where $g \in \mathbb{F}_{2^n}$ and $e = 2^t - 1$ for some $t \geq 2$, the classical algorithm being the square-and-multiply algorithm, cf. [MOV96, Sect. 14.6]. For this algorithm, the number of multiplications requested for this exponent is equal to the number of squares $t - 1$.

This algorithm can be modified in order to minimize the total number of multiplications. In particular, consider Algorithm 9, which is a slight variation of the original algorithm. In order to compute x^{2^t-1}, the number of multiplications for the previous algorithm is $\lceil t/2 \rceil$, while the number of squares is t. As a result, if one cares only of the total number of multiplications[7], this algorithm is better than the original one (at the cost to pre-compute and store the quantity $g^2 \cdot g$). Thus, using the previous analysis about the number of rounds, the total number of multiplications $\mathcal{M}$ and of squares $\mathcal{S}$ for MiMC-n/n are given by

$$\mathcal{M} = \left\lceil \frac{t}{2} \right\rceil \cdot \left\lceil \frac{\log_2 n}{\log_2(2^t - 1)} \right\rceil \qquad \mathcal{S} = t \cdot \left\lceil \frac{\log_2 n}{\log_2(2^t - 1)} \right\rceil.$$

To give a concrete example, for $n = 129$, the best result is obtained for $t = 4$ (that is for the exponent 15)[8], for which the total number of multiplications is 66 (instead of 82 for the exponent 3), while the number of squares is 99 (instead of 82 for the exponent 3).

[6] Only if the polynomial is *not* sparse and it has degree d, this number of terms corresponds to $d + 1$.

[7] For completeness, we mention that other variants of such algorithm can be more competitive - regarding the number of multiplications - for large t. For our purpose, we limit ourselves to consider the algorithm just given, and we refer to [AGR+16, Sect. 5.3] for more details.

[8] Actually, the best result is obtained for $t = 6$, that is for the exponent 63. However, since $\gcd(63, 2^{129} - 1) = 7$, the round function defined using the exponent 63 is not a permutation.

```
Data: $g \in \mathbb{F}_{2^n}$ and $e = 2^t - 1$ for some $t \geq 2$
Result: $g^e$
$g_0 \leftarrow g$;
$g_1 \leftarrow g^2 \cdot g$;
$A \leftarrow 1$; for each $i$ from $0$ to $\lfloor t/2 \rfloor$ do
    $A \leftarrow (A^2)^2$;
    $A \leftarrow A \cdot g_1$;
end
if $t$ mod $2 \neq 0$ then
    $A \leftarrow A^2$;
    $A \leftarrow A \cdot g_0$;
end
return $A$
```

Algorithm 9: Modular exponentiation with cache.

To conclude, if the cost of a square operation is negligible with respect to the cost of a multiplication (that is, if the square operation is linear), then it is possible to minimize the total number of multiplications choosing an exponent[9] of the form $2^t - 1$ different from 3. Instead, when the number of square operations can not be ignored (as for example in the case of SNARK settings or in the $GF(p)$ case), the choice of an exponent of the form 2^t-1 different from 3 does not offer any advantage due to the fact that the total number of multiplications and square operations

$$\mathcal{M} + \mathcal{S} = \left(\left\lceil \frac{t}{2} \right\rceil + t \right) \cdot \left\lceil \frac{\log_2 n}{\log_2(2^t - 1)} \right\rceil \approx \frac{3}{2} \log_2 n$$

is almost constant. Since we propose MiMC for different applications, the exponent 3 seems to be the best optimal choice.

Finally, only for completeness, it is also possible to extend the previous analysis to the case $GF(p)$ (where $p \approx 2^n$). However, we remark that in this case each square operation counts as a multiplication, since it is not linear. Thus, if we consider an exponent of the form $2^t - 1$, the total number of multiplications M for MiMC-p/p is

$$\mathcal{M} = \left(\left\lceil \frac{t}{2} \right\rceil + t \right) \cdot \left\lceil \frac{\log_2(\log_2 p)}{\log_2(2^t - 1)} \right\rceil.$$

Also in this case, the exponent 3 turns out to be the optimal choice.

Remark - Computation Cost Model. In most models of computation, a field multiplication is considered to be computationally more expensive than an addition. However, note that squaring is a linear operation in a binary field $GF(2^n)$. Hence, if we consider the number of non-linear multiplications in a binary field, then the number required to compute x^3 is just one.

However, this does not hold in the SNARK setting, where each witness variable – and possibly each constraint – is generated from a field operation (more specifically from a field multiplication). As a consequence, computing x^3 generates two equations $x \cdot x = y$ and $y \cdot x = x^3$. Hence, in this setting we do not benefit from the linearity of squaring over the fields $\mathbb{F}_{2^n}$ and computing x^3 costs two multiplications. On the other hands, the cost of additions in these fields is still negligible compared to that of multiplication. Note that we can also disregard the cost of multiplication by a constant.

Finally, we stress that *although the cost of an addition is considered negligible compared to a multiplication, very large number of additions can reduce the efficiency of a design.*

[9] We remark that we did not consider statistical attacks in the previous analysis. If one chooses an exponent different than 3, one may also considers and includes a security analysis against statistical attacks.

10.4. Application

Remark. *Since I did not work on the practical applications/implementations of MiMC, I limit myself to recall here the main results and I refer to [AGR+16] and [GRR+16] for a detailed discussion on it. The results of this section are due to the work of Arnab Roy (SNARKs applications) and Dragos Rotaru, Peter Scholl and Nigel P. Smart (MPC applications) respectively.*

10.4.1. SNARKs Applications

The main idea of the SNARK is to provide a circuit whose satisfiability enables a verifier to check correctness of an underlying computation. In our concrete implementation, we focus on the (zk)SNARK for arithmetic circuit satisfiability.

The main target of our design proposals is to improve the efficiency of (zk)SNARK when they are used as cryptographic primitives in a SNARK setting. Due to a lack of an alternative, SHA-256 has been used for various applications in verifiable computing [CFH+15] and applications of SNARKS like Zerocash [BCG+14], which leads to a bottleneck in efficiency.

Benchmarking Environment. For all field operations we used the NTL library together with the gf2x library. All computations were carried out on an Intel Core i7 2.10GHz processor with 16GB memory and we took the average over ≈ 2000 repetitions.

Results. As we demonstrate in [AGR+16], a sponge construction instantiated by a MiMC permutation compares very favorably in (zk)SNARKs applications. Based on our experiments and implementations, we report a factor 10 improvement w.r.t. SHA-256. For processing a single block i.e. for hashing a single block message our MiMC implementation in the SNARK setting requires ≈ 7.8 milliseconds to generate the arithmetic circuit and witness while SHA-256 takes ≈ 73 milliseconds.

Since LowMC was designed for MPC/ZK applications we have also implemented it in the SNARK setting. A comparison of LowMC with MiMC is given in Table 10.1. As a result, based on our experiments and implementations, we report approximately a factor 12 improvement w.r.t. LowMC.

Table 10.1.: Comparison of LowMC and MiMC with block size 1025 and the corresponding parameters for LowMC. Number of rounds and number of Sboxes per round are denoted as R and m respectively.

	MiMC	**LowMC**	
		$R = 16$	$R = 55$
		$m = 196$	$m = 20$
total time	7.8ms	90.3ms	271.2ms
constraint generation	6.3ms	13.5ms	9.2ms
witness generation	1.5ms	76.8ms	262.0ms
# addition	646	8420888	28894643
# multiplication	1293	9408	3300
# rank-1 constraint	646	4704	2200

As a design with an unusual imbalance between ANDs and XORs, the comparison with LowMC variants is interesting as it gives an example where the number multiplications alone can no longer be used as a hint for the eventual performance. Where the round-minimized LowMC variant is more than 10 times slower with about 8 times more multiplications, reducing the number of ANDs in the other LowMC variant at the expense of many more rounds does not have the expected effect: the runtime grows again. The reason is the huge amount of XOR computations whose cost is clearly are no longer negligible. This shows the limits of a simplified metric that focuses on AND gates (or multiplication gates) also.

Conclusion. In conclusion, our analysis shows that due to the designed balance between addition and multiplication over higher field MiMC achieves more efficiency than well known classic design like SHA-2 as well as recently proposed LowMC. Results of our analyses suggest that although multiplication in a larger field is more expensive than multiplication in lower field, an enormous number of field addition in a lower field can reduce the efficiency of a design targeted for certain ZK-applications like SNARK.

10.4.2. MPC Applications

In [GRR+16], we conduct a study of some PRFs for use in Multi-Party Computation, including also new protocols for evaluating number-theoretic PRFs and implementations of “traditional” block cipher candidates designed to have a low complexity in MPC. Our focus is on secret sharing based MPC systems such as that typified by BDOZ [BDOZ11], SPDZ [DPSZ12], and VIFF [DGKN09], or any classical protocol based on Shamir Secret Sharing. In such situations data is often shared as elements of a finite field $\mathbb{F}_p$ of large prime characteristic.

To better understand this scenario, a simple example application of MPC is to enable distributed secure storage of long-term cryptographic keys, by secret-sharing the key and storing each share at a separate server. When the key is required by an application such as encryption or authentication, the MPC protocol is used to compute this functionality. If this cryptographic functionality is a symmetric cipher, then this application would be greatly enhanced by using an “MPC-Friendly” symmetric primitive.

Benchmarking Environment. In any application of MPC, one of the most important factors affecting performance is the capability of the network. We ran benchmarks in a standard 1Gbps LAN setting, and also a simulated WAN setting, which restricts bandwidth to 50Mbps and latency to 100ms, using the Linux tc tool. This models a real-world environment where the parties may be in different countries or continents. In both cases, the test machines used have Intel i7-3770 CPUs running at 3.1GHz, with 32GB of RAM.

Results. The results of our experiments in the LAN and WAN environments are shown in the following Tables, respectively. All the result are the average of 5 experiments, each of which ran at least 1000 PRF operations.

LowMC obtains slightly better throughput and latency than AES over a LAN. In the WAN setting, LowMC gets a very high throughput of over 300 blocks per second. This is due to the low online communication cost for multiplications in $\mathbb{F}_2$ instead of $\mathbb{F}_{2^n}$ or $\mathbb{F}_p$, and the fact that local computation is less significant in a WAN.

In both scenarios, the Legendre PRF gives the lowest latency, even when outputing 128-bit field elements rather than bits, due to its low round complexity. The single-bit output variant achieves by far the highest throughput of all the PRFs, so would be ideally suited to an application based on a short-output PRF. The Legendre PRF with large outputs is useful in scenarios where low latency is very important, although the preprocessing costs are expensive compared to MiMC below.

The Naor-Reingold PRF also achieves a low latency - though not as good as the Legendre PRF - but it suffers greatly when it comes to throughput.

Table 10.2.: Two-party performances of several PRFs in a LAN setting (where "op(s)" ≡ operation(s))

PRF	Best Latency	Best Throughput		Preproc
	(ms/op)	(Batch Size)	(ops/s)	(ops/ms)
AES	7.713	2048	530	5.097
LowMC ("vector")	4.302	256	591	2.562
LowMC ("M4R")	4.148	64	475	2.565
NR(128) ("log")	4.375	1024	370	4.787
NR(128) ("const")	4.549	256	281	2.384
Leg ("bit")	0.349	2048	202969	1225
Leg ("1")	1.218	128	1535	9.574
MiMC ("basic")	12.007	2048	8788	33.575
MiMC ("cubic")	5.889	1024	6388	33.575

Table 10.3.: Two-party performances of several PRFs in a WAN setting (where "op(s)" ≡ operation(s))

PRF	Best Latency	Best Throughput		Preproc
	(ms/op)	(Batch Size)	(ops/s)	(ops/ms)
AES	2640	1024	31.947	0.256
LowMC ("vector")	1315	2048	365	0.1259
LowMC ("M4R")	659	2048	334	0.1261
NR(128) ("log")	713	1024	59.703	0.2359
NR(128) ("const")	478	1024	30.384	0.1175
Leg ("bit")	202	1024	2053	60.241
Leg ("1")	210	512	68.413	0.4706
MiMC ("basic")	7379	512	59.04	1.650
MiMC ("cubic")	3691	512	79.66	1.650

Leg ≡ Legendre PRF - NR ≡ Naor-Reingold PRF

The MiMC cipher seems to provide a good compromise amongst all the prime field candidates, especially as it also performs well when performed "in the clear". The "cube" variant[10], which halves the number of rounds, effectively halves the latency compared to the naive protocol. This results in a slightly worse throughput in the LAN setting due to the higher computation costs, whereas in the WAN setting round complexity is more important. Although the latency is much higher than Legendre PRFs, due to the large number of rounds, MiMC achieves the best throughput for $\mathbb{F}_p$-bit outputs, with over 6000 operations per second. In addition, the pre-processing costs of MiMC are better than that of both Legendre and the Naor-Reingold PRFs.

Conclusion. In conclusion, one would likely prefer the Legendre PRF for applications which require low latency, and which do not involve any party external to the MPC engine, and MiMC for all other applications.

10.4.3. Other Applications

For completeness, we mention that other people coming up with more use-cases of MiMC.

[10] We consider two different approaches for computing MiMC in MPC, with a secret shared key and message. The "basic" approach is simplest, whilst the "cube" variant has half the number of rounds of communication, with slightly more computation. More details can be found in [GRR+16, Sect. 5.2].

Verifiable Delay Functions. A "verifiable delay function" (VDF) [BBBF18] requires a *specified* number of sequential steps to evaluate, yet produces a unique output that can be efficiently and publicly verified. VDFs have many applications in decentralized systems, including public randomness beacons, leader election in consensus protocols, and proofs of replication.

In [BBBF18], authors present new candidate constructions that are the first to achieve an exponential gap between evaluation and verification time. In particular, a theoretical VDF can be constructed using incrementally verifiable computation (IVC) [Val08]. IVC can be constructed from succinct non-interactive arguments of knowledge (SNARKs) under a suitable extractor complexity assumption. For this particular case, MiMC turns out to be the one of the best possible choices since *(1st)* it is a "SNARK friendly" hash function (or permutation) over $\mathbb{F}_p$ and *(2nd)* the decryption process of MiMC is much more "expensive" (e.g. in terms of non-linear operations) than the corresponding encryption process.

Modes of Operation Suitable for Computing on Encrypted Data. In [RSS17], authors examine how two parallel modes of operation for Authenticated Encryption (namely CTR+PMAC and OTR mode) work when evaluated in a multi-party computation engine. These two modes are selected because they suit the PRFs examined in previous works, they are highly parallel, and do not require evaluation of the inverse of the underlying PRF.

They examine how the currently best PRFs for secret shared MPC over $\mathbb{F}_p$, namely MiMC and Leg, can be used to enable nonce-based authenticated encryption,where we benchmark a number of orthogonal options. Without going into the details, it turns out that that MiMC will often significantly outperform Leg.

STARKs. As briefly recalled, ZK-SNARKs - succinct zero knowledge proof technology - that can be used for all sorts of use-cases ranging from verifiable computation to privacy-preserving cryptocurrency. W.r.t. ZK-SNARKs, ZK-STARKs [BBHR18] resolves one of the primary weaknesses of ZK-SNARKs, its reliance on a "trusted setup" (T stands indeed for "transparent"). ZK-STARK is the first realization of a transparent ZK system in which verification scales exponentially faster than database size, and moreover, this exponential speedup in verification is observed concretely for meaningful and sequential computations, described next.

A (concrete) practical implementation of ZK-STARKs has be done/proposed using MiMC[11].

[11] See `https://vitalik.ca/general/2018/07/21/starks_part_3.html` for more details.

11 Feistel MiMC and GMiMC

As we highlighted in the previous chapter, one drawback of MiMC is that the decryption process is much more expensive that the encryption one. A simple way to fix this problem is to turn the MiMC Even-Mansour cipher into a Feistel one, since in this last case, the encryption process and the decryption one are identical expect for the order of the round keys and round constants.

As we are going to show, the possibility to set up competitive Meet-in-the-Middle attacks on Feistel MiMC require (approximately) to double the number of rounds with respect to MiMC in order to guarantee the same security for Feistel MiMC. As a result, it seems that Feistel MiMC can not be competitive for the applications that we have in mind (where the goal is to minimize the number of multiplications). Thus, the only advantage of the Feistel approach seems to be that decryption is as cheap as an encryption computation.

In [AGP+18], we show that this conclusion does not hold for Generalized Feistel constructions [Nyb96]. In particular, our analysis suggests that for unbalanced Feistel schemes with an expanding round function we do not have to increase the number of rounds further for $t > 2$ branches – – up to a certain *finite* limit $t \leq t^\star$ – compared to $t = 2$ branches considered in [AGR+16]. For practical use cases, we show that a high number of branches can be meaningful, hence allowing for an up to 100-fold improvement of multiplication-related metrics, which influence the performance of applications which depend on metrics like “the number of multiplications” or “the product of field size × number of multiplications” or similar, compared to MiMC.

The new block cipher that we are going to present based on unbalanced Feistel schemes is called “Generalized MiMC”, GMiMC for simplicity. As we are going to show, main applications of GMiMC are MPC applications, PQ-signature schemes (based on zero-knowledge protocols) and SNARKs applications. In particular, whereas MiMC was not competitive at all in a recently proposed new class of PQ-secure signature schemes, our new construction leads to about 30 times smaller signatures than MiMC. In MPC use cases, where MiMC outperforms all other competitors, we observe improvements in throughput by a factor of more than 7 and simultaneously a 16-fold reduction of preprocessing effort, albeit at the cost of a higher latency. Another use case, where MiMC already outperforms other designs (i.e. in the area of SNARKs), sees modest improvements. Additionally, this use case benefits from the flexibility to use smaller fields.

11.1. Description of Feistel and Generalized MiMC

Notation. In a Feistel network with $t \geq 2$ branches, X_{i-1} denotes the input to the branch i, where $1 \leq i \leq t$. X_{t-1} and X_0 denote the inputs to the leftmost and rightmost branches respectively. $X_i \in \mathbb{F}$ for a finite field $\mathbb{F}$. The block size (in bits) of the keyed Feistel permutation is denoted by N, while $n = \lceil \log_2 |\mathbb{F}| \rceil$ denotes the branch size (in bits). We write $\mathbb{F}_p$ for the finite prime field of order p. We write $\mathbb{F}_{2^q}$ for any finite field of order 2^q. The bit size of the key of a block cipher is denoted by κ. In particular, through the paper we work with two different cases (depending on the practical implementation), denoted as:

- the *univariate* case, for which the key-size is $\kappa = n = \lceil \log_2 |\mathbb{F}| \rceil$;
- the *multivariate* case, for which the key-size is $\kappa = N = n \cdot t = \lceil \log_2 |\mathbb{F}| \rceil \cdot t$.

11.1.1. Feistel MiMC

Working in $\mathbb{F}_{2^n}$, FeistelMiMC-$2n/\kappa$ is similar to MiMC-n/n, with the only exception that the Even-Mansour construction is replaced by a Feistel Network where the round consists of three steps:

- a key addition with a key k (of size n bit);
- the addition of a round constant $c_i \in \mathbb{F}_{2^n}$;
- the application of a non-linear function defined as $F(x) := x^3$ for $x \in \mathbb{F}_{2^n}$.

The ciphertext is finally produced by adding the key k again to the output of the last round. Hence, the round function is described as

$$x_L \leftarrow x_R, \qquad x_R \leftarrow (x_R \oplus k \oplus c_i)^3 \oplus x_L$$

where $c_0 = c_r = 0$, while the other constants are chosen at random in $\mathbb{F}_{2^n}$. In the last round, the swap operation is not applied. Finally, with respect to MiMC, it is not required that the cubic function is invertible.

The key-schedule is discussed in the following. As we are going to show, the number of rounds to provide security is given by

$$r = 2\lceil \log_3 2 \cdot n \rceil + 1 \qquad \text{and} \qquad r = 2\lceil \log_3 2 \cdot n \rceil + 3$$

respectively for the cases $\kappa = n$ and $\kappa = N = 2n$.

In an analogous way, FeistelMiMC-$2p/\kappa$ is similar to MiMC-p/p working in $\mathbb{F}_p$.

11.1.2. The Block Cipher GMiMC

We construct generalized MiMC (GMiMC) variants from several generalized (unbalanced and balanced) Feistel networks, e.g. with contracting round function (CRF), expanding round function (ERF), Nyberg's GFN and a new structure which we call Multi-Rotating (MR). Each of the following constructions[1] is a keyed permutation over $\mathbb{F}_{2^n}$ or $\mathbb{F}_p$. The three main parameters of the block ciphers are denoted by $[\kappa, t, n]$. For example, GMiMC$_{crf}[4n, 4, n]$ denotes the permutation GMiMC with CRF which has branch size n, key size $4n$ and number of branches 4. The numbers of rounds for all constructions are given in Table 11.3. The key-schedule is linear and equal for all the proposed designs, and it is discussed in the following. All round constants are chosen randomly and fixed.

GMiMC$_{\mathbf{crf}}$

An unbalanced Feistel network (UFN) with a contracting round function (CRF) can be written as

$$(X_{t-1}, X_{t-2}, \dots, X_0) \leftarrow (X_{t-2}, X_{t-3}, \dots, X_{t-1} + F(X_{t-2}, \dots, X_0))$$

where X_i is the input to the i-th branch of the Feistel network and $F(\cdot)$ is a key-dependent function in round j, cf. Figure 11.1. In GMiMC$_{crf}$ we define the j-th round function as

$$F(x_0, \dots, x_{t-2}) := \left(\sum_i x_i + k_j + c_j \right)^3$$

where c_j are distinct round constants (for $1 \leq j \leq r$) and k_j is the key to the round j.

[1] The following construction are defined over $\mathbb{F}_p$. The description in the case $\mathbb{F}_{2^n}$ is equivalent by replacing the sum $+$ with the XOR-sum $\oplus$.

[2] ***Acknowledgement.*** *Figure 11.1 – made by Arnab Roy – has been copied from [AGP+18].*

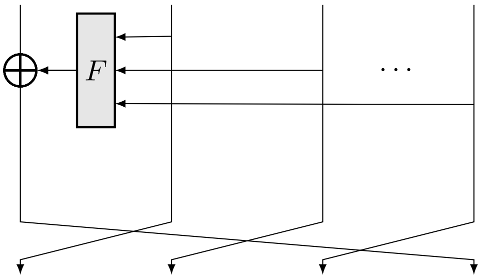


Figure 11.1.: One round of a t-branch unbalanced Feistel network (UFN) with a contracting round function (CRF).[2]

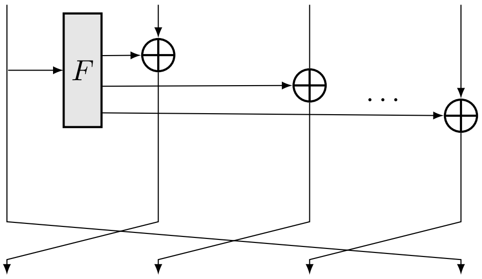


Figure 11.2.: One round of a t-branch unbalanced Feistel network (UFN) with an expanding round function (ERF).[3]

GMiMC$_{\mathbf{erf}}$

An unbalanced Feistel network with an expanding round function (ERF) can be written as

$$(X_{t-1}, X_{t-2}, \ldots, X_0) \leftarrow (X_{t-2} + F(X_{t-1}), \ldots, X_0 + F(X_{t-1}), X_{t-1})$$

where X_i is the input to the i-th branch of the Feistel network and $F(\cdot)$ is a key-dependent function in round j, cf. Figure 11.2. In GMiMC$_{erf}$ the j-th round function is defined as

$$F(x) \coloneqq (x + k_j + c_j)^3$$

where k_j and c_j are as in GMiMC$_{crf}$.

GMiMC$_{\mathbf{Nyb}}$

A generalized Feistel network was proposed in [Nyb96] for an even number of branches and can be written as

$$(X_{t-1}, \ldots, X_1, X_0) \leftarrow \big(X_0, X_{t-1} + F_0(X_0), X_{t-2} + F_1(X_1), \ldots, X_{t/2+1}, \ldots, X_1\big)$$

Each $F_i(\cdot)$ in the j-th round of GMiMC$_{Nyb}$ is defined as

$$F_i(x) \coloneqq \big(x + k_{i+j\cdot t/2} + c_{i+j\cdot t/2}\big)^3,$$

where $c_{i+j\cdot t/2}$ are distinct constants in round j and $k_{i+j\cdot t/2}$ are round keys, cf. Figure 11.3.

[3] ***Acknowledgement.*** *Figure 11.2 – made by Arnab Roy – has been copied from [AGP+18].*

[4] ***Acknowledgement.*** *Figure 11.3 – made by Markus Schofnegger – has been copied from [AGP+18].*

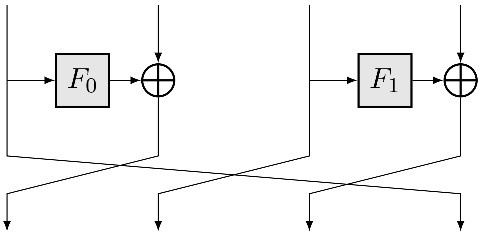


Figure 11.3.: One round of an 8-branch Nyberg Generalized Feistel Network (GFN)[4]

Remark - GMiMC$_{\mathbf{mrf}}$. *In [AGP+18], we also propose GMiMC$_{mrf}$, that is a Multi-Rotating Feistel network that provides extremely fast diffusion. The first such variants were proposed in [SM10] and later implemented in the block cipher Twine [SMMK12]. In those cases, the simple rotation of the branches used between the calls to the Feistel functions is replaced with a more sophisticated permutation. In our case, the branch permutation is replaced by a simple rotation applied on half of the branches. When this rotation is the same in every round, this structure is reminiscent of a type-II generalized Feistel network. Thus, our second idea consists in changing this rotation at every round. Since this design has been proposed and analyzed by Leo Perrin, I refer to the paper for more details, and I omit it from this Thesis.*

Key Schedule

When $\kappa = n$ (i.e. the univariate case), then $k_i = k$ for each i. The key-schedule for the multivariate case $\kappa = t \times n$ is a little more complicated. Let $k = k_0||k_1||\ldots||k_{t-1}$, and let M be a $t \times t$ matrix with elements in $GF(2^n)$ or $GF(p)$ that satisfies the following condition:

- M is invertible[5], that is there exists M^{-1};
- for each $1 \le i \le \lceil R/t \rceil$ where R is the number of rounds, then[6]

  $$M^i[j,l] \equiv (\underbrace{M \times M \times .... \times M}_{i\text{-th times}})[j,l] \neq 0$$

  for all $0 \le j, l < t$, where $X[j,l]$ denotes the coefficient in row j and column l of the matrix X.

For each $1 \le i \le \lceil R/t \rceil$ let

$$[k_{i\cdot t}||k_{i\cdot t+1}||\ldots||k_{(i+1)\cdot t-2}||k_{(i+1)\cdot t-1}]^T = M \times [k_{(i-1)\cdot t}||k_{(i-1)\cdot t+1}||\ldots||k_{i\cdot t-2}||k_{i\cdot t-1}]^T.$$

The second condition on M guarantees that *each subkey k_j for $j > t$ - linearly - depends on all the first t subkeys.* This fact has an important consequence. Consider GMiMC$_{crf}$ and/or GMiMC$_{erf}$ instantiated with a key schedule that uses the sub-keys cyclically, i.e. $k_{i,j} = \hat{k}_{j\cdot t/2+i \pmod t}$. If the attacker guesses $t-1$ subkeys, then she can *potentially* skip both the first and the last $t-1$ rounds. Instead, in the case in which each subkey - linearly - depends on all the first t subkeys, this strategy simply does not apply. As a result, the proposed key-schedule allows to save a certain number of rounds (approximately $t-1$) w.r.t. a key schedule that uses the sub-keys cyclically. Similar argumentation - but on a smaller scale - holds for GMiMC$_{Nyb}$.

[5] Let A be a lower triangular matrix and let B be an upper triangular matrix. Then $M = B \times A$ is invertible.

[6] If no matrix exists that satisfies the following condition, then one must choose a matrix M for which the total number of zero coefficients for each M^i is minimum.

Remark - Round Constants. As for MiMC, we remark that also the key-schedule of GMiMC consists of a round-constant addition. This is "hidden" in the definition of each round function $F_i(\cdot)$, e.g. $F_i(\cdot) = (\cdot + k + c_i)^3$ for a random round-constant c_i. We highlight that it is possible to define an equivalent key-schedule where the round-constant addition is already included in the key-schedule, e.g. $\hat{k}_i := k_i + c_i$ where k_i is defined by the previous key-schedule.

11.1.3. Hash Function

To construct the hash function GMiMCHash, we use one of the previous structures, e.g. the GMiMC$_{erf}$, with fixed sub-keys[7], e.g. $0^{n \cdot R}$, where R is the number of rounds. Denoting the fixed key permutation as GMiMC$^{\pi}_{\text{erf}}[\kappa, t, n]$, GMiMCHash is constructed by instantiating a sponge construction [BDPA07] with GMiMC$^{\pi}_{\text{erf}}[\kappa, t, n]$. The number of rounds of the permutation GMiMC is chosen according to Table 11.3 - *univariate* case $2^{\kappa} \simeq 2^n \simeq p$.

When the internal permutation $\mathcal{P}$ of an N-bit sponge function (composed of c-bit capacity and r-bit bitrate – $N = c + r$) is modeled as a randomly chosen permutation, it has been proven by Bertoni *et al.* [BDPA08] to be indifferentiable from a random oracle up to $2^{c/2}$ calls to $\mathcal{P}$. In other words, a sponge with a capacity of c provides $2^{c/2}$ collision and $2^{c/2}$ (second) preimage resistance. Given a permutation of size N and a desired security level s, we can hash $r = N - 2s$ bits per call to the permutation.

As usual, the message is first padded according to the sponge specification so that the number of message blocks is a multiple of r, where r is the rate in sponge mode. For GMiMCHash-l we use a GMiMC permutation where $N = n \cdot t = 4 \cdot l + 1$ and $s = 2 \cdot l$. For GMiMCHash-256 we thus use a GMiMC permutation with $N = n \cdot t = 1024$ or 1025. The rate and the capacity are chosen as 512 and 513 respectively. This choice allows for processing the same amount of input bits as SHA-256 (512 bits) while at the same time offering collision security and (second) preimage security of 256 bits. We highlight that while we could use any of the GMiMC constructions, GMiMC$_{\text{erf}}$ turns out to be the most efficient choice in several settings as shown in Section 11.6.2.

11.2. Security Analysis

As for any new design, it is paramount to present a concrete security analysis. In the following, we provide an in-depth analysis of the security of the GMiMC family of block ciphers. In particular, for each proposal we consider the maximum number of rounds that can be attacked by the attacks currently present in the literature.

Important Remark. *Due to our target applications, here we limit ourselves to provide the number of rounds to guarantee security *only* in the following two scenarios:*

- *GMiMC instantiated over* $\mathbb{F}_p$ (used for applications like SNARKs and MPC);
- *GMiMC instantiated over* $\mathbb{F}_{2^n}$ *in the low-data scenario* (used for application like PQ-Signature Scheme).

We stress that this choice is motivated by the fact that we focus on the scenario that are useful for our applications. Thus, even if GMiMC can be instantiated over $\mathbb{F}_{2^n}$*, we do not provide the number of rounds to guarantee security in this scenario.*

Before going on, we remark that *many (almost all) attacks work in the same way in* $\mathbb{F}_p$ *and in* $\mathbb{F}_{2^n}$*. One of the few exception to this fact is the higher-order differential attack.* More details on this are given in the following.

[7] We emphasize that no key-schedule is required in this case, since there is no secret-key material.

Table 11.1.: Minimum number of rounds required to provide security against the corresponding attacks when $2^\kappa \simeq 2^n \simeq p$ - no restriction on data complexity - and $t > 2$. For simplicity, $2 \cdot \log_3(2) = 1.262$.

	GMiMC$_{crf}$	GMiMC$_{erf}$	GMiMC$_{Nyb}$
GCD	$\lceil 1.262 \cdot \log_2(p) - 4\log_3(\log_2(p))\rceil + 2t$	$\lceil 1.262 \cdot \log_2(p) - 4 \cdot \log_3(\log_2(p))\rceil + 2t - 2$	$\lceil 1.262 \cdot \log_2(p) - 4\log_3(\log_2(p))\rceil + t + 2$
Interpolation	$\lceil 1.262 \cdot \log_2(p)\rceil + 4t - 3$	$\lceil 1.262 \cdot \log_2(p)\rceil + 2t$	$\lceil 1.262 \cdot \log_2(p)\rceil + t + 2$
Higher Order (in $\mathbb{F}_p$)	$2 + 4t + 2\log_3(t)$	$2 + 2t + 2\log_3(t)$	$2 + t + 2\log_3(t)$
(Trunc.) Differential	$2 + (t^2 + t) \cdot \lceil \frac{\log_2(p)}{2(\log_2(p)-1)} \rceil$	$2 + (t^2 + t) \cdot \lceil \frac{\log_2(p)}{2(\log_2(p)-1)} \rceil$	$3t + 2$
Impossible Diff.	$3t - 1$	$2t$	$2t$

Table 11.2.: Minimum number of rounds required to guarantee the security against the corresponding attacks when $2^\kappa \simeq 2^N \simeq p^t$ - no restriction on data complexity - and $t > 2$. For simplicity, $2 \cdot \log_3(2) = 1.262$.

	GMiMC$_{crf}$	GMiMC$_{erf}$	GMiMC$_{Nyb}$
Guess + GCD	$\lceil 1.262 \cdot \log_2(p) - 4\log_3(\log_2(p))\rceil + 3t - 1$	$\lceil 1.262 \cdot \log_2(p) - 4 \cdot \log_3(\log_2(p))\rceil + 3t - 3$	$\lceil 1.262 \cdot \log_2(p) - 4\log_3(\log_2(p))\rceil + t + 3$
Interpolation	$\lceil 1.262 \cdot \log_2(p)\rceil + 5t - 4$	$\lceil 1.262 \cdot \log_2(p)\rceil + 3t - 2$	$\lceil 1.262 \cdot \log_2(p)\rceil + t + 3$
Gröbner Basis	$\lceil 0.631 \cdot \log_2(p) + 2\log_3(t)\rceil + 4t - 3$	$\lceil 0.631 \cdot \log_2(p) + 2\log_3(t)\rceil + 4t - 5$	$\lceil 0.631 \cdot \log_2(p) + 2\log_3(t)\rceil + t + 2$
Higher Order (in $\mathbb{F}_p$)	$1 + 5t + 2\log_3(t)$	$1 + 3t + 2\log_3(t)$	$3 + t + 2\log_3(t)$
(Trunc.) Differential	$1 + t + (t^2 + t) \cdot \lceil \frac{\log_2(p)}{2(\log_2(p)-1)} \rceil$	$1 + t + (t^2 + t) \cdot \lceil \frac{\log_2(p)}{2(\log_2(p)-1)} \rceil$	$3t + 3$
Impossible Diff.	$4t - 2$	$3t - 1$	$2t + 1$
"Generic" Attack	$5t - 3$	$4t - 2$	-

Security Analysis – GMiMC instantiated over $\mathbb{F}_p$. Almost all the attacks are independent of the fact whether (a) the size of the key is equal to the branch size $\kappa = n$ (equivalently, $2^\kappa \simeq p$ for the $\mathbb{F}_p$ case) or (b) equal to $\kappa = N = t \cdot n$ (equivalently, $2^\kappa \simeq p^t$ for the $\mathbb{F}_p$ case). Table 11.1 and Table 11.2 summarize the minimum number of rounds required to guarantee the security against several possible attacks respectively in the first and in the second case - we assume $t > 2$ in both cases. The number of rounds of GMiMC is then chosen in order to provide security to all possible attack vectors.

Starting from the results proposed in the following section, in Section 11.5 we list the minimum number of rounds for each construction, together with some useful observations for the possible applications like MPC, SNARKs and post-quantum signature schemes.

Note: given the number of rounds of a distinguisher that is independent of the secret key, we decided to add 2 rounds - in order to prevent *key-guessing attack* - for the univariate case. For the multivariate case, we decided to add $(t + 1)$ rounds - in order to prevent *key-guessing attack* - for GMiMC$_{\text{crf}}$ and GMiMC$_{\text{erf}}$, and 3 rounds - in order to prevent *key-guessing attack* - for GMiMC$_{\text{Nyb}}$. This choice is supported by the definition of the key-schedule, in particular by the fact that each subkey depends linearly on all the first t subkeys (we refer to the previous section for details).

Security Analysis – GMiMCHash instantiated over $\mathbb{F}_p$. For the hash function GMiMCHash case, the number of rounds of the inner permutation is chosen according to the corresponding univariate case (referring to Table 11.1). This is due to the following considerations. First, as we just recalled in the previous section, when the internal permutation $\mathcal{P}$ of an $N = c + r$ bit sponge function is modeled as a randomly chosen permutation, the sponge hash function is indifferentiable from a random oracle up to $2^{c/2}$ calls to $\mathcal{P}$. The numbers of rounds of the univariate case is sufficient to guarantee security against any (secret-/known-/chosen-) distinguisher which is independent of the key. Equivalently, this means that such number of rounds guarantee that $\mathcal{P}$ does not present any non-random/structural property (among the ones known in the literature[8]). It follows that the

[8] That is, we do not exclude that a non-random property can be discovered in the future.

previous assumption is satisfied. These and the fact that every key-recovery attack is meaningless in the hash scenario support our choice to consider the univariate case in order to determine the number of rounds of the inner permutation.

Before going on, we remark that the fact that $\mathcal{P}$ presents a non-random/structural property does not imply an attack on the hash sponge function instantiated by $\mathcal{P}$. To have a concrete example, consider Keccak (SHA-3). A zero-sum distinguisher can be set up for the *full* 24-round internal permutation that defines it - see for example [BCC11]. In other words, the internal permutation that defines Keccak presents a non-random property, that is it does not look like a pseudo-random permutation[9]. On the other hands, the best practical collision attack covers ("only") up to 6-round Keccak [QSLG17], which is still far from threatening the security of the full 24-round Keccak family.

Security Analysis – GMiMC instantiated over $\mathbb{F}_{2^n}$ in the *low-data scenario.* For some practical applications considered in the following, we also consider the case in which the attacker has a limited access to data (e.g. 1 or 2 (plaintext, ciphertext) pairs). The security analysis for this particular case is proposed in Sect. 11.4. As we are going to show, due to the fact that the attacker can have access to few (plaintext, ciphertext) pairs, only few attacks (e.g. the GCD one) apply to this case. We remark that all the attacks that we are going to consider in this scenario work in the same way in $\mathbb{F}_{2^n}$ and $\mathbb{F}_p$. As a result, we mainly re-use the results proposed in the Sect. 11.3.

11.3. Security Analysis – GMiMC instantiated over $\mathbb{F}_p$

11.3.1. Algebraic Attacks

In this section, we consider algebraic attacks against Feistel MiMC and GMiMC. These attacks are particularly relevant for the applications where the attacker has access only to a limited number of (plaintext, ciphertext) pairs available to the attacker.

Greatest Common Divisors

As for the original MiMC [AGR+16], an attack strategy is to compute the Greatest Common Divisors (GCD). In particular, given more than one known (plaintext, ciphertext) pair or working on the output of different branches of a single known (plaintext, ciphertext) pair (as described in the following), one can construct their polynomial representations and compute their polynomial GCD to recover a multiple of the key[10]. Note that this is a known-plaintext attack, and not a chosen-plaintext one.

Since interpolation attack is more efficient than GCD attack (from the attacker point of view), we refer to Sect. 11.4 for all details about GCD attack, while we refer to Table 11.1 for the minimum number of rounds that ensure security against the GCD attack.

Before going on, we remark that this is one of the few attacks that applies in the low-data scenario, considered in one of the following applications (i.e. post-quantum signatures). More details on this fact are given in the following.

Gröbner Bases

The natural generalization of GCDs to the multivariate case is the notion of a Gröbner basis [BKW93]. The attack proceeds like the GCD attack with the final GCD computation replaced by a Gröbner basis computation. Analogous to the GCD analysis, we highlight that the Feistel structure permits to construct multivariate "meet-in-the-middle" polynomials, we denote their degree as d_i in this

[9] We also refer to [BDPA] for a detailed discussion about this topic.

[10] Improving the computational complexity of this attack using more pairs is an open problem. However, since the cost is dominated by the size of the polynomials involved, it is not clear that significant improvements are possible.

subsection and define $d = \min_i d_i$.

Complexity. For generic systems, the complexity of computing a Gröbner basis for a system of $\mathfrak{N}$ polynomials in $\mathfrak{V}$ variables is

$$\mathcal{O}\left(\binom{\mathfrak{V} + D_{reg}}{D_{reg}}^{\omega}\right) \tag{11.1}$$

operations over the base field $\mathbb{F}$ [BFP12], where D_{reg} is the *degree of regularity* and $2 \leq \omega < 3$ is the linear algebra constant. We note that the memory requirement of these algorithms is of the same order as running time. The degree of regularity depends on the degrees of the polynomials d and the number of polynomials $\mathfrak{N}$. When $\mathfrak{V} = \mathfrak{N}$, we have a simple closed form [BFSY05]

$$D_{reg} = 1 + \sum_{i=0}^{\mathfrak{N}-1} (d_i - 1),$$

where d_i is the degree of the i-th polynomial f_i in the polynomial system we are trying to solve. In the over-determined case, i.e., $\mathfrak{V} < \mathfrak{N}$, the degree of regularity can be estimated by developing the Hilbert series of an ideal generated by generic polynomials $\langle f_0, \dots, f_{\mathfrak{N}-1}\rangle$ of degrees d_i. We stress that this analysis presumes that the polynomials considered here behave like generic systems, which is in accordance with our practical experiments. Closed form formulas for D_{reg} are known for some special cases, but not in general.

In particular, each plaintext/ciphertext pair – denoted by $p, c \in (\mathbb{F}_{2^n})^t$ where $p \equiv (p_0, ..., p_{t-1})$ and $c \equiv (c_0, ..., c_{t-1})$ – gives a system of t equations

$$\forall i = 0, ..., t-1 : \qquad c_i = f_i(p_0, ..., p_{t-1}, k_0, ..., k_{t-1})$$

in t variables $k_0, ..., k_{t-1}$ (note that the key-schedule is linear), where f_i are functions of degree d.

The introduction of new intermediate variables to reduce the degree of the involved polynomials does not lead to a reduced solving time as this increases the number of variables with $\mathfrak{V}$. On the other hand, depending on parameter choices, the hybrid approach [BF09; BFP12] which combines exhaustive search with Gröbner basis computations may lead to a somewhat reduced cost. Following [BF09; BFP12], guessing $\varphi \leq \mathfrak{V}$ components of the key leads to a complexity of

$$\mathcal{O}\left(p^{\varphi} \cdot \binom{\mathfrak{V} - \varphi + D'_{reg}}{D'_{reg}}^{\omega}\right),$$

where $D'_{reg} \leq D_{reg}$ is the degree of regularity for the system of equation after substituting φ variables with their guesses.

Noting, though, that guessing a variable in a monomial reduces its degree and that guesses only affect a subset of rows in the Macaulay matrix, we will more conservatively assume an overall cost of

$$\mathcal{O}\left(p^{\varphi} \cdot \binom{\mathfrak{V} - \varphi + D'_{reg} - 1}{D'_{reg} - 1}^{\omega}\right). \tag{11.2}$$

Many known pairs. Each new (plaintext, ciphertext) pair provides a new polynomial while keeping the number of unknowns $\mathfrak{V} = t$ constant. Thus, given that there are $\binom{t+d}{d}$ monomials of degree less than or equal to d in t unknowns, we may simply collect $\binom{t+d}{d}$ polynomials from the same number of known (plaintext, ciphertext) pairs. In this over-determined case (that is, number of equations n_e bigger than number of variables n_v), there is no closed formula to compute D_{reg}. By definition, the degree of regularity is defined as the index of the first non-positive coefficient in

$$H(z) = \frac{\prod_{i=1}^{n_e}(1 - z^{d_i})}{(1-z)^{n_v}} = \frac{(1 - z^{3^r})^{n_e}}{(1-z)^{n_v}} = (1 - z^{3^r})^{n_e - n_v} \cdot (1 + z + z^2)^{n_v},$$

where n_e is the number of equations, n_v is the number of variables, and $d_i = 3^r$ is the degree of the i-th equation. By simple observation, the index of the first non-positive coefficient can not be smaller than $d = 3^r$, since $(1 + z + z^2)^{n_v}$ contains only positive terms. Thus, the overall complexity becomes $\mathcal{O}(\binom{t+d}{d}^{\omega})$ with the hidden constant ≥ 1. Following the method above, we expect $D'_{reg} = D_{reg} - 1 = d - 1$ for the hybrid approach. Plugging the (MiMT) degrees from the previous sections into d then produces the expected overall solving time.

GMiMC$_{\textbf{crf}}$ *(case: $2^\kappa = p^t$).* To prevent the Gröbner basis attack, the minimum number of rounds r must satisfy

$$p^{\varphi} \cdot \binom{t - \varphi + d - 1}{d - 1}^{\omega} \geq p^t,$$

for all $\varphi \in \{0, \ldots, t-2\}$ and where the degree d is a function of the number of rounds r, that is, $d = d(r)$.

A key datum is the degree reached in each of our constructions after r rounds. Consider the t-branch GMiMC$_{crf}$ and denote the branches by $(X_{t-1}, \ldots, X_2, X_1, X_0)$. Given a plaintext, the degrees of the t keys growth differently in the t multivariate polynomials corresponding to the t branches of the Feistel network. In round r, the polynomial of the leftmost branch has the least degrees, which are given by

$$d_{i,j} = \begin{cases} 3^{r-(i-1)-j} & \text{if } r > i + j - 1, \\ 0 & \text{otherwise.} \end{cases}, \tag{11.3}$$

where $j = t - 1$ denotes the leftmost branch and the degree of variable k_i in branch j is $d_{i,j}$. For all (algebraic) attacks in the following, we only care of the minimum degree:

$$d_{t-1,t-1} = \min_i \min_j d_{i,j} = 3^{r-2t+2},$$

where $0 \leq i \leq t-1$ and $0 \leq j < t$. For the follow-up, we remark that the degree of each word of the plaintext in the t-branch is given by formula (11.3) both for the univariate and multivariate case.

For our parameter choices, this expression is minimized for $\varphi = 0$. We thus require

$$\binom{t+d}{d}^{\omega} = \binom{t + 3^{r-2t+2}}{3^{r-2t+2}}^{\omega} \approx p^t.$$

By simple computation, we get

$$\binom{t+d}{d} = \frac{1}{t!} \cdot \prod_{i=1}^{t} (d+i) \geq \frac{d^t}{t!} \geq \left(\frac{d}{t}\right)^t = 2^{t \log_2(d/t)}$$

where $n! \leq n^n$ for each $n \geq 1$, and, setting $\omega = 2$, we obtain

$$2t \log_2(d/t) \approx n \cdot t.$$

In our case:

$$2t \log_2(d/t) = 2t \log_2(3^{r-2t+2}/t) \approx \log_2(p)\, t \text{ or } r = \left\lceil 2t + \frac{\log_3 2}{2} \log_2(p) - 2 + \log_3 t \right\rceil.$$

To thwart Meet-in-the-Middle attacks, this value is doubled.

To conclude, we emphasize that we use $d(r) = 3^{r-2t+2}$ in order to compute the previous number of rounds. Since 3^{r-2t+2} is the minimum of the degrees of the variables, it is plausible that a lower number of rounds is sufficient to protect against Gröbner basis attacks. Also, we reiterate that these attacks require roughly the same amount of memory as elementary operations. The same consideration holds for the other ciphers of the GMiMC family.

GMiMC$_{\mathbf{erf}}$ *(case: $2^\kappa = p^t$).* After $r \geq t$ rounds[11], the minimum degree of a variable in the output polynomials is 3^{r-t}.

To prevent the Gröbner basis attack, we require

$$\binom{t+d}{d}^{\omega} = \binom{t+3^{r-t}}{3^{r-t}}^{\omega} \approx p^t.$$

Working as before, we obtain

$$2t\log_2(d/t) = 2t\log_2(3^{r-t}/t) \approx t \cdot \log_2(p) \text{ or } r = \left\lceil t + \frac{\log_3 2}{2}\log_2(p) + \log_3 t \right\rceil.$$

Further Consideration. In order to compute the final number of rounds for GMiMC$_{erf}$, one must take care of a variant of the previous attack. Let X_i^r be the output of the i-th branch after r rounds. Assume $t \geq 3$ and consider the output of two branches, e.g. the output of the branches in position 1 - denoted by X_1^r - and 2 - denoted by X_2^r. By definition

$$X_i^r = X_{i-1}^{r-1} \oplus (X_t^{r-1} \oplus k \oplus c)^3,$$

where $i = 1, 2$, k is the secret key (remember that we are working in the case $\kappa = n$) and c is the round constant. Note that X_j^s is a function of the key k, that is, $X_j^s = X_j^s(k)$. It is simple to observe that

$$X_1^r \oplus X_2^r = X_0^{r-1} \oplus (X_t^{r-1} \oplus k \oplus c)^3 \oplus X_1^{r-1} \oplus (X_t^{r-1} \oplus k \oplus c)^3 = X_0^{r-1} \oplus X_1^{r-1},$$

that is, $X_1^r \oplus X_2^r$ is still a function of k, but *the degree of such a function is lower than the degree of the functions that define X_1^r and X_2^r.*

Since this same trick can be repeated multiple times[12], in order to prevent this attack it is sufficient to increment the number of rounds by $t-3$. As a result, the minimum number of rounds is approximately given by

$$r = \left\lceil 2t + \frac{\log_3 2}{2}\log_2(p) + \log_3 t - 3 \right\rceil.$$

To thwart Meet-in-the-Middle attacks, this value is doubled.

GMiMC$_{\mathbf{Nyb}}$ *(case: $2^\kappa = p^t$).* To prevent the Gröbner basis attack, we require

$$\binom{t+d}{d}^{\omega} = \binom{t+3^{r-t/2}}{3^{r-t/2}}^{\omega} \approx p^t.$$

Working as before, we obtain

$$2t\log_2(d/t) = 2t\log_2(3^{r-t/2}/t) \approx t \cdot \log_2(p) \text{ or } r = \left\lceil t/2 + \frac{\log_3 2}{2}\log_2(p) + \log_3 t \right\rceil.$$

To thwart Meet-in-the-Middle attacks, this value is doubled.

[11] For our goal, we do not need all the details regarding the degree for $r < t$.

[12] For completeness, we mention another possible strategy to prevent this attack. Instead of incrementing the number of rounds, one possibility is to use a different constant for each branch of each round. In other words, consider GMiMC$_{erf}$ as defined in Section 11.1.2 for the case $\kappa = n$ (a similar argument holds also for the case $\kappa = t \cdot n$). The expanding round function (ERF) can be re-written as

$$(X_{t-1}^{(j+1)}, X_{t-2}^{(j+1)}, \ldots, X_0^{(j+1)}) \leftarrow (X_{t-2}^{(j)} + F_{t-2}(X_{t-1}^{(j)}), \ldots, X_0^{(j)} + F_0(X_{t-1}^{(j)}), X_{t-1}^{(j)}),$$

where the round function is defined as

$$F_i(x) := (x + k + c_j)^3,$$

and where the random constants c_i are different for each branch.

This strategy allows to prevent the given attack without increasing the number of rounds. On the other hand, since our final goal is to minimize the total number of multiplications, this strategy is less efficient than the one proposed in the main text. Indeed, let r' be the number of rounds necessary to prevent the attack. The strategy proposed in the main text requires $r' + (t-3)$ multiplications, while the one just given requires $r' \cdot (t-1)$ multiplications, where $r' + (t-3) < r' \cdot (t-1)$ for each $t \geq 3$ (and $r' \geq 1$).

Interpolation Attack

As for the original MiMC, one of the most powerful attacks against the GMiMC family is the interpolation attack, introduced by Jakobsen and Knudsen [JK97] in 1997. The strategy of the attack is to construct a polynomial corresponding to the encryption function without knowledge of the secret key. If an adversary can construct such a polynomial then for any given plaintext the corresponding ciphertext can be produced without knowledge of the secret key.

Let $E_k : \mathbb{F}_q \to \mathbb{F}_q$ be an encryption function. For a randomly fixed key k, the polynomial $P(x)$ representing $E_k(x)$ – where x is the indeterminate corresponding to the plaintext – can be constructed using the *Vandermonde matrix*[13] - cost approximately of $\mathcal{O}(m^2)$ - or the *Lagrange's theorem*[14] - cost approximately of $\mathcal{O}(m \cdot \log m)$, where m is the number of monomials of $P(\cdot)$.

This method can be extended to a key-recovery attack. The attack proceeds by simply guessing the key of the final round, decrypting the ciphertexts and constructing the polynomial for $r-1$ rounds. With one extra (plaintext, ciphertext) pair, the attacker checks whether the polynomial is correct.

Each output branch of a (balanced or unbalanced) Feistel network can be represented as a multivariate polynomial where the variables are the inputs to each branch. If the maximum degree of a single variate monomial in one of these output polynomials is low, then an attacker can exploit this property to mount an attack on the block cipher.

Using this idea, we first briefly describe at high level generic attack(s) on the GMiMC block ciphers, focusing on a t-branch Feistel network. Let us denote the t input branches as $x_{t-1}, ..., x_1$ and x_0 from left to right. Suppose the polynomials over the field representing the output branches are denoted by $P_i \in \mathbb{F}_{2^n}[X]$ $(i = 0, 1, ..., t-1)$ and d_i denotes the degree of the polynomial P_i. Working as in [JK97], the number of monomials of such polynomial is well approximated by $\prod_{i=0}^{t-1}(d_i + 1)$. It follows that if the condition

$$\prod_{i=0}^{t-1}(d_i + 1) \approx 2^N \simeq p^t$$

is fulfilled, then the attacker requires the full code-book in order to construct the interpolation polynomial[15]. As a result, such polynomial can not be used for a key-recovery attack or for a forgery attack.

GMiMC$_{\mathbf{crf}}$. As we have just seen in (11.3), the minimum degree of the output polynomials for each branch (after r rounds) is lower bounded by 3^{r-2t+2}. Due to the previous discussion, GMiMC$_{\text{crf}}$ is secure against interpolation attack if

$$(3^{r-2t+2})^t \approx 2^N \simeq p^t.$$

Hence, $r = \frac{\log_2(p)}{\log_2 3} + (2t-2)$ rounds will be secure against the above-mentioned attacks. Conservatively, $2r+2$ rounds will be secure against meet-in-the-middle attacks/distinguishers for the case $2^\kappa \simeq p$, while $2r+t+1$ rounds will be secure against meet-in-the-middle attacks/distinguishers for the case $2^\kappa \simeq p^t$.

[13] Given the interpolation polynomial $P(x) = a_t x^t + a_{t-1}x^{t-1} + ... + a_2 x^2 + a_1 x + a_0$, it interpolates the data points (x_i, y_i) in the sense that $P(x_i) = y_i$ for all $i \in \{0, 1, ..., t\}$, where $E_k(x_i) = y_i$ for $i = 0, 1, ...t$. By substituting the first equation in here, one gets a system of linear equations in the coefficients a k. By solving this system for a k, one can construct the interpolant polynomial $P(x)$. If one re-writes this system in a matrix-vector form, the matrix defined by the terms $\{x_{j,i}\}_{0\le i,j\le t}$ is commonly referred to as a Vandermonde matrix, and the cost to invert a $(t+1)\times(t+1)$ Vandermonde matrix (and so to construct the interpolation polynomial) is $\mathcal{O}(t^2)$.

[14] If the polynomial has degree d, we can find it using Lagrange's formula $P(x) = \sum_{i=0}^{d} y_i \cdot \prod_{0\le j\le d, i\neq j} \frac{x-x_j}{x_i-x_j}$, where $E_k(x_i) = y_i$ for $i = 0, 1, \ldots d$.

[15] *Remark.* Due to the cost of constructing the interpolation polynomial (approximately $\mathcal{O}(m \log m)$ where m is the number of monomials), we emphasize that the cost of such attack is higher than the cost of a brute-force attack if condition (11.3.1) is satisfied.

GMiMC$_{\text{erf}}$. Working as in Sect. 11.3.1, the minimum degree of the output polynomials for each branch is lower bounded by $3^{r-(t-1)}$ (after $r \geq t$ rounds). Due to the argumentation proposed in Sect. 11.3.1, GMiMC$_{\text{erf}}$ is secure against interpolation attack if

$$(3^{r-(t-1)})^t \approx 2^N \simeq p^t.$$

Hence, $r \approx \frac{\log_2(p)}{\log_2 3} + (t-1)$ rounds will be secure against the above-mentioned attacks. Conservatively, $2r+2$ rounds will be secure against meet-in-the-middle attacks/distinguishers for the case $2^\kappa \simeq p$, while $2r+t+1$ rounds will be secure against meet-in-the-middle attacks/distinguishers for the case $2^\kappa \simeq p^t$.

GMiMC$_{\text{Nyb}}$. Let $t = 2 \cdot t'$. We use a set of (plaintext, ciphertext) pairs to do the interpolation analysis. Working as before, after $r > 2$ rounds, the minimum degree of the output polynomials will be $3^{r-1} = d_j$ for some branch j (even j). In order to get the maximum degree $3^{r-1} = p$, the number of rounds must satisfy $r \approx \frac{\log_2(p)}{\log_2 3} + 1$. For securing the cipher against MITM-type attacks/distinguishers, we use $2r$ rounds. Finally, we add t rounds to provide full diffusion and avoid key-guessing.

In the case $2^\kappa = p^t$, we have to add 1 more round in order to prevent the combination of the interpolation attack and the brute-force one.

Remark - Interpolation Attack and Hash Functions. One may ask if a similar attack is meaningful in the hash scenario (where there is no key or/and secret material). Here we briefly show a concrete example.

Given the inner permutation $\mathcal{P}$ of a sponge construction, assume that it is possible to construct the interpolation polynomial *without* using the full code-book. In this case, such a polynomial can be exploited to set up a *forgery attack* on the permutation $\mathcal{P}$, which is instead not possible for a (pseudo-)random permutation. As a result, the inner permutation $\mathcal{P}$ of the sponge construction can be distinguish from a (pseudo-)random permutation, which means that the sponge hash function is not indifferentiable from a random oracle (as showed in [BDPA08] and recalled in Sect. 11.1.3).

In conclusion, in order to avoid such a distinguisher, it is sufficient that the number of rounds of the inner permutation GMiMC - instantiated with a fixed key - of the sponge construction is equal to the number of rounds necessary to prevent the interpolation attack discussed here (equivalently, the number of rounds necessary to ensure that the internal permutation has maximum degree).

Higher-Order Differential Attack

Let $\mathcal{A}$ be an affine space. Higher-order differential attacks [Knu94] exploit the fact that $\bigoplus_{x\in\mathcal{A}} P(x) = 0$ if the dimension of $\mathcal{A}$ is higher than the degree of $P(\cdot)$. In other words, a higher-order differential attack can be mounted by choosing an affine space — like $\mathcal{A}$ — of dimension $d+1$ (or, equivalently, of size 2^{d+1}) if P has degree at most d. To thwart higher-order differential attacks, the number of rounds must be chosen in order to ensure that the algebraic degree of the GMiMC family of block ciphers is bigger than the biggest subspace in $\mathbb{F}$.

Higher-Order Differential in $\mathbb{F}_p$ *versus* Higher-Order Differential in $\mathbb{F}_{2^N}$. Due to the strategy exploited by the higher-order differential attack, there is a crucial difference between the cases $\mathbb{F}_{2^N}$ and $\mathbb{F}_p$.

As we have just seen, given a function $f(\cdot)$ of degree d, the sum over the outputs of the function applied to all elements of a vector space $\mathcal{V}$ of dimension $\geq d+1$ is zero. *The crucial point here is that the previous result holds if $\mathcal{V}$ is a (sub)space, and not only a generic set of elements.* While $\mathbb{F}_{2^m}$ is always a subspace of $\mathbb{F}_{2^n}$ for each $m \leq n$, the only subspaces of $\mathbb{F}_p$ are $\{0\}$ and $\mathbb{F}_p$. It follows that the biggest subspace of $\big(\mathbb{F}_p\big)^t$ has dimension t, with respect to the biggest subspace of $\big(\mathbb{F}_{2^n}\big)^t$, which has dimension $n \cdot t = N$.

This fact has an important impact on the higher-order differential attack: *if a cipher is instantiated over $\mathbb{F}_p$, then a lower degree (and hence a smaller number of rounds) is sufficient to protect it from the higher-order differential attack with respect to the number of rounds required for the $\mathbb{F}_{2^N}$ case.* In particular, it is sufficient that both the encryption and the decryption functions[16] have degree at most t, with respect to degree N of the case of $\mathbb{F}_p$.

Higher-Order Differential on GMiMC$_{\textbf{crf}}$ instantiated over $\mathbb{F}_p$. Due to the analysis proposed in Sect. 11.3.1, the minimum degree of GMiMC$_{\text{crf}}$ after $r > 2t - 1$ rounds is (at least) 3^{r-2t}. The condition $3^{r-2t} \geq t$ is satisfied by $r \geq 2t + \log_3(t)$. Finally, we add 2 rounds in order to avoid key-guessing attack for the univariate case and $t + 1$ rounds for the multivariate case.

Higher-Order Differential on GMiMC$_{\textbf{erf}}$ instantiated over $\mathbb{F}_p$. Using the same analysis proposed before, the minimum degree of GMiMC$_{\text{erf}}$ after $r > t$ rounds is (at least) 3^{r-t}. The condition $3^{r-t} \geq t$ is satisfied by $r \geq t + \log_3(t)$. In order to avoid distinguishers on GMiMCHash, we simply double this number of rounds. Finally, we add 2 rounds in order to avoid key-guessing attack for the univariate case and $t + 1$ rounds for the multivariate case.

Higher-Order Differential on GMiMC$_{\textbf{Nyb}}$ instantiated over $\mathbb{F}_p$. Using the same analysis proposed before, the minimum degree of GMiMC$_{\text{Nyb}}$ after $r > t$ rounds is (at least) 3^{r-1}. The condition $3^{r-1} \geq t$ is satisfied by $r \geq 1 + \log_3(t)$. In order to avoid distinguishers on GMiMCHash, we simply double this number of rounds. Finally, we add t rounds in order to avoid key-guessing attack for the univariate case and in order to provide full diffusion. One more round is added for the multivariate case.

Higher-Order Differential on GMiMC instantiated over $\mathbb{F}_{2^n}$ – Some Remarks. Since we do not require GMiMC instantiated over $\mathbb{F}_{2^n}$ for our target applications, we stress that we do not claim anything about the minimum number of rounds necessary to protect GMiMC w.r.t. an Higher-Order Differential over $\mathbb{F}_{2^n}$. In any case, we briefly discuss this case, and we highlight the main open problem that one has to face when considering this attack.

In order to choose the number of rounds, one has to *estimate the growth of the degree.* First of all, since the degree of the round function in its algebraic representation in $\mathbb{F}_{2^n}$ is only 2, the algebraic degree of one round is 2 as well. Clearly, the algebraic degree of the cipher after r rounds is bounded from above by 2^r. However, a better and more realistic upper bound can be evaluated by using the division property [Tod15b], introduced by Todo at Eurocrypt 2015. As a main result, it turns out that the degree of the function – when it is iterated – grows in a much smoother way than expected when it approaches the number of variables. For instance, the degree of the composition of two functions $G \circ F(\cdot)$ can always be upper-bounded by $\deg(G \circ F) \leq \deg(G) \cdot \deg(F)$. This trivial bound, however, is often very little representative of the true degree of the permutation, in particular if we are trying to estimate the degree after a high number of rounds. An analogous result for SPN ciphers was previously found by e.g. Boura *et al.* [BCC11].

While the (just cited) results proposed by Boura *et al.* work for SPN ciphers, no equivalent results is given in the literature for Feistel constructions. Moreover, division property is a useful tool to study the growth of the degree when one considers a single cipher instantiated by fixed parameters n and t (or a "small" number of them), but it does not provide a generic formula that can work for any possible choice of parameters n and t. However, due to the scope of this work, the choice of the parameters depends on the performance of the practical applications, and it cannot be done in

[16] Note that the attacker works at word level (i.e. with element of $\mathbb{F}_p$) in the case of an higher order differential attack instantiated over $\mathbb{F}_p$. Instead, for the case $\mathbb{F}_{2^n}$, the attacker can work both at word level (i.e. with element of $\mathbb{F}_{2^n}$) or at bit level (i.e. with element of $\mathbb{F}_2$).

advance. In conclusion, a future open problem would be to determine a *tight* bound for the growth of the degree for a generic Feistel construction, as the one provided in [BCC11].

Zero-Sum Partitions and Sponge GMiMCHash. Here we briefly discuss how to apply the previous analysis in the case of a sponge construction instantiated by one of the GMiMC structures, e.g. the GMiMC$_{erf}$, with a fixed key, e.g. 0^κ. Since the key is fixed, the previous key-recovery attacks are meaningless. On the other hand, previous analysis about the degree of GMiMC can be applied also in this scenario.

As showed in [BDPA08] and recalled in Sect. 11.1.3, when the internal permutation $\mathcal{P}$ of a sponge function is modeled as a randomly chosen permutation, then the sponge construction is indifferentiable from a random oracle up to $2^{c/2}$ calls to $\mathcal{P}$.

A possible distinguisher that can be set up in order to distinguish the inner permutation GMiMC from a (pseudo-)random one is the one based on the *zero-sum partition*.

Definition 24 (Zero-sum Partition [BCC11])**.** *Let P be a permutation from $\mathbb{F}_{2^n}$ to $\mathbb{F}_{2^n}$. A zero-sum partition for P of size $K = 2^k \lneqq 2^n$ is a collection of 2^k disjoint sets $\{X_1, X_2, ..., X_k\}$ sets with the following properties:*

- *$X_i = \{x^i_1, ..., x^i_{2^{n-k}}\} \subset \mathbb{F}_{2^n}$ for each $i = 1, ..., k$ and $\bigcup_{i=1}^{2^{n-k}} X_i = \mathbb{F}_{2^n}$;*
- *for each $i = 1, ..., 2^k$, the set X_i satisfies zero-sum*

$$\bigoplus_{j=1}^{2^k} x^i_j = \bigoplus_{j=1}^{2^k} P(x^i_j) = 0.$$

A similar definition works also in the case of $(\mathbb{F}_p)^t$. As we have just seen, if f is a k-degree function on $\mathbb{F}_{2^n}$, then $\bigoplus_{v \in V \oplus a} f(v) = 0$ for any $(k+1)$-dimension subspace $V \subseteq \mathbb{F}_{2^n}$, where $V \oplus a$ is an arbitrary coset of V.

To avoid this distinguisher, it is sufficient that the number of rounds of the inner permutation GMiMC - instantiated with a fixed key - of the sponge construction is equal to the number of rounds necessary to prevent the higher-order differential attack discussed in the previous section. If this is not the case, a *zero-sum partition* can be mounted. For completeness, we recall that, while it is known how to construct a zero-sum[17] for a random permutation (see [AM; BDPA] for details), there is no way - to the best of our knowledge - to construct a zero-sum partition for a random permutation without using a brute-force approach. In conclusion, it follows that the assumption "the internal permutation GMiMC is indifferentiable from a random oracle" does not hold e.g. in the case in which GMiMC is instantiated with a lower number of rounds than the one determined by the higher-order differential attack.

We remark that a similar approach based on zero-sum partitions is largely used in the literature to set up attack or to investigate the security of sponge hash functions (see e.g. Keccak [AM; BCC11], PHOTON [WGR18], . . .).

11.3.2. Statistical Attacks

Here we consider statistical attacks against GMiMC. All statistical attacks that we are going to analyze work in the same way both for the case in which GMiMC is instantiated over $\mathbb{F}_p$ or/and over $\mathbb{F}_{2^n}$. For this reason, in the following we do *not* study separately the two scenario.

[17] We remark that for a zero-sum, it is sufficient to find a *single* set Z of inputs z_i for which $\bigoplus_i z_i = \bigoplus_i P(z_i) = 0$.

Classical and Truncated Differential Cryptanalysis

Differential cryptanalysis [BS90; BS93] and its variations are the most widely used techniques to analyze symmetric-key primitives. The differential probability of any function over the finite field $\mathbb{F}_{2^n}$ is defined as

$$\Pr[\alpha \to \beta] := |\{x : f(x) + f(x + \alpha) = \beta\}|/2^n.$$

It is well known that the function $f(x) = x^3$ is *Almost Perfect Non-linear (APN)* [NK92] and, thus, has optimal differential probability over a prime field or $\mathbb{F}_{2^n}$. For this function the probability is bounded above by $2/2^n$ or $2/|\mathbb{F}|$. In the following, we provide the minimum number of rounds to guarantee security against this attack. A variant of classical differential cryptanalysis is the truncated differential one [Knu94], in which the attacker can predict only part of the difference between pairs of texts.

As largely done in the literature, we assume that the cipher is secure against differential attack if any (truncated) differential characteristic has probability lower than 2^{-N}.

GMiMC$_{\mathbf{crf}}$. In order to find the minimum number of rounds to protect the cipher against differential attack, we look for the best possible (truncated) differential characteristic. Consider an input difference of the form $(0, \ldots, 0, \Delta_I, \Delta_I)$ where $\Delta_I \neq 0$. It is straightforward to observe that such input difference does not active any S-Box in the first $r_0 = t - 2$ rounds (since the input difference is always zero), that is the output difference after r_0 rounds is $(\Delta_I, \Delta_I, 0, \ldots, 0)$. After $r_1 = t - 1$ round, we get an output difference of the form $(\Delta_I, 0, ..., 0, \Delta_I \oplus f^{r_1}(\Delta_I))$, where $f^{r_1}(\cdot)$ denotes the r_1-th round function. Observe that $\Delta_I \oplus f^{r_1}(\Delta_I) = 0$ with prob. 2^{-n+1}. Indeed[18], since an active (cubic) S-Box maps its non-zero input difference to 2^{n-1} possible output differences each one with prob. 2^{-n+1}, it follows that $f^{r_1}(\Delta_I) = \Delta_I$ with probability 2^{-n+1}. Assume $f^{r_1}(\Delta_I) = \Delta_I$. After $r_2 = t$ rounds, we get an output difference of the form $(0, ..., 0, \Delta_I)$, while after $r_3 = t + 1$ rounds, we get an output difference of the form $(0, ..., 0, \Delta_I, f^{r_3}(\Delta_I))$. Due to the previous consideration, $f^{r_3}(\Delta_I) = \Delta_I$ with prob. 2^{-n+1}.

As a result, the following (truncated) characteristic over $t + 1$ rounds

$$(0, \ldots, 0, \Delta_I, \Delta_I) \xrightarrow[\text{prob. } 1]{R^{t-2}(\cdot)} (\Delta_I, \Delta_I, 0, \ldots, 0) \xrightarrow[\text{prob.}\leq 2^{-n+1}]{R(\cdot)} (\Delta_I, 0, \ldots, 0) \xrightarrow[\text{prob. } 1]{R(\cdot)}$$
$$\xrightarrow[\text{prob. } 1]{R(\cdot)} (0, \ldots, 0, \Delta_I) \xrightarrow[\text{prob. } \leq 2^{-n+1}]{R(\cdot)} (0, \ldots, 0, \Delta_I, \Delta_I)$$

has an overall probability equal to 2^{-2n+2}. Before going on, note that any other input difference active at least one S-Box in the first $t - 2$ rounds. In other words, it seems not possible to find a longer (truncated) characteristic with lower probability.

By iterating this (truncated) characteristic, it is possible to construct a (truncated) differential characteristic over $s \cdot (t + 1)$ with probability at most $(2^{-2n+2})^s$. By simple computation, $(2^{-2n+2})^s \leq 2^{-N}$ if and only if $(2n - 2) \cdot s \geq N$, that is $s \geq \lceil \frac{N}{2n-2} \rceil$. As a result, $2 + t \cdot (t + 1) \cdot \lceil \frac{n}{2(n-1)} \rceil$ rounds are sufficient to provide security in the univariate case, while $1 + t + t \cdot (t + 1) \cdot \lceil \frac{n}{2(n-1)} \rceil$ rounds are sufficient to provide security in the multivariate case.

GMiMC$_{\mathbf{erf}}$. Working in the same way as before, consider an input difference of the form $(0, \ldots, 0, \Delta_I)$ where $\Delta_I \neq 0$. It is straightforward to observe that such input difference does not active any S-Box in the first $r_0 = t - 1$ rounds (since the input difference is always zero), that is the output difference after r_0 rounds is $(\Delta_I, 0, 0, \ldots, 0)$. After $r_1 = t$ round, we get an output difference

[18] Note that the cubic S-Box is APN. This means that for each non-zero input/output difference, the number of solutions of S-Box$(x \oplus \delta_I) \oplus$ S-Box$(x) = \delta_O$ is at most 2. As a result, for each δ_I there are at most $2^n/2$ different δ_O for which the previous equation as at least one solution.

of the form $(f^{r_1}(\Delta_I), ..., f^{r_1}(\Delta_I), \Delta_I)$, where $f^{r_1}(\cdot)$ denotes the r_1-th round function. Observe that $\Delta_I = f^{r_1}(\Delta_I)$ with prob. 2^{-n+1}. Indeed, since an active (cubic) S-Box maps its non-zero input difference to 2^{n-1} possible output differences each one with prob. 2^{-n+1}, it follows that $f^{r_1}(\Delta_I) = \Delta_I$ with probability 2^{-n+1}. Assume $f^{r_1}(\Delta_I) = \Delta_I$, that is an output difference of the form $(\Delta_I, ..., \Delta_I)$. After $r_2 = t+1$ rounds, we get an output difference of the form $(\Delta_I \oplus f^{r_2}(\Delta_I), ..., \Delta_I \oplus f^{r_2}(\Delta_I), \Delta_I)$. Due to the previous consideration, $f^{r_2}(\Delta_I) = \Delta_I$ with prob. 2^{-n+1}.

As a result, the following (truncated) characteristic over $t+1$ rounds

$$(0, \ldots, 0, 0, \Delta_I) \xrightarrow[\text{prob. } 1]{R^{t-1}(\cdot)} (\Delta_I, 0, 0, \ldots, 0) \xrightarrow[\text{prob.} \leq 2^{-n+1}]{R(\cdot)} (\Delta_I, \Delta_I, \ldots, \Delta_I) \xrightarrow[\text{prob. } \leq 2^{-n+1}]{R(\cdot)} (0, \ldots, 0, \Delta_I)$$

has an overall probability equal to 2^{-2n+2}. Before going on, note that any other input difference active at least one S-Box in the first $t-1$ rounds. In other words, it seems not possible to find a longer characteristic with lower probability.

By iterating this (truncated) characteristic, it is possible to construct a differential characteristic over $s \cdot (t+1)$ with probability at most $(2^{-2n+2})^s$. By simple computation, $(2^{-2n+2})^s \leq 2^{-N}$ if and only if $(2n-2) \cdot s \geq N$, that is $s \geq \lceil \frac{N}{2n-2} \rceil$. As a result, $2 + t \cdot (t+1) \cdot \lceil \frac{n}{2(n-1)} \rceil$ rounds are sufficient to provide security in the univariate case, while $1 + t + t \cdot (t+1) \cdot \lceil \frac{n}{2(n-1)} \rceil$ rounds are sufficient to provide security in the multivariate case.

GMiMC$_{\textbf{Nyb}}$. First of all, note that any input difference of the form $(0, \Delta_I, 0, \ldots, \Delta_I)$, where $\Delta_I \neq 0$, does not activate any S-Box in the first round. Working as in the previous case, it is possible to prove that the following (truncated) characteristic over $3t/2 = 3t'$ rounds (where $t = 2t'$)

$$(\Delta, 0, \ldots, 0) \xrightarrow{R^{3t'}(\cdot)} (\Delta', 0, \ldots, 0)$$

has probability $2^{(t-1)\cdot(-n+1)}$, where in general $\Delta \neq \Delta'$.

By iterating this characteristic, it is possible to construct a differential characteristic over $s \cdot (3t')$ with probability at most $(2^{(t-1)\cdot(-n+1)})^s$. As a result, $(2^{(t-1)\cdot(-n+1)})^s \leq 2^{-N}$ if and only if $(t-1)\cdot(n-1)\cdot s \geq 2N$, that is $s \geq 2$ (since $N \geq 2(n+t-1)$ due to the fact that $n \cdot (t-2) \geq 2(t-1)$ for each $n \geq 3$). As a result, $2 + 3t$ rounds are sufficient to provide security in the univariate case, while $3 + 3t$ rounds are sufficient to provide security in the multivariate case.

Impossible Differential Cryptanalysis

Impossible differential cryptanalysis was introduced by Biham *et al.* [BBS99] and Knudsen [Knu98]. This cryptanalytic technique exploits differentials occurring with probability 0. It has been very successful against FNs and led to the best cryptanalysis against well known FN-based block ciphers like CLEFIA and CAMELLIA [BNS14].

The approach used in the following - and largely exploited in the literature - to construct impossible differential is to combine two (truncated) differentials with prob. 1 that collide in the middle.

GMiMC$_{\textbf{crf}}$. As first thing we look for a probability-one truncated differential in order to construct impossible differentials for GMiMC$_{crf}$. A probability-one differential for a maximum of $t-1$ rounds of this UFN with t branches is described as follows:

$$(0, \ldots, 0, \alpha, \alpha) \to (0, \ldots, 0, \alpha, \alpha, 0) \to \ldots \to (\alpha, \alpha, 0, \ldots, 0).$$

A truncated differential with probability 1 exists for $(t-1)+(t-1) = 2t-2$ rounds. This is described as follows:

$$(0, \ldots, 0, \alpha, \alpha) \xrightarrow{t-1\, rounds} (\alpha, \alpha, 0, \ldots, 0) \xrightarrow{t-1\, rounds} (0, *, \ldots, *).$$

This will allow us to attack $3t-3$ rounds of the cipher, exploiting the differential just given on $2t-2$ rounds and noting that $(\beta, 0, \ldots, 0) \xrightarrow{t-1\, rounds} (0, *, \ldots, *)$ with probability 1. As a result, the $(3t-3)$-rounds impossible differential used for the attack is given by

$$(0, \ldots, 0, \alpha) \xrightarrow[\text{prob. } 1]{R^{2t-2}(\cdot)} (0, *, \ldots, *) \neq (\beta, 0, \ldots, 0) \xleftarrow[\text{prob. } 1]{R^{t-1}(\cdot)} (0, *, \ldots, *)$$

for $\alpha, \beta \neq 0$. Hence, the number of iterations to protect the cipher against such an attack must be at least $[(2t-2)+(t-1)]+2 = 3t-1$ for the case $\kappa = n$. For the case $\kappa = t \cdot n$, the number of rounds must be at least $[(2t-2)+(t-1)]+t+1 = 4t-2$.

GMiMC$_{\text{erf}}$. A probability-one differential exists for a maximum of $t-1$ rounds of the cipher, which is given as follows:

$$(0, \ldots, 0, \alpha) \to (0, \ldots, 0, \alpha, 0) \to \ldots (\alpha, 0, \ldots, 0).$$

This differential can be extended to a probability-one truncated differential for t rounds as follows:

$$(0, \ldots, 0, \alpha) \xrightarrow{t-1\, rounds} (\alpha, 0, \ldots, 0) \xrightarrow{1\, round} (*, *, \ldots, *, \alpha).$$

This probability-one differential allows us to construct an impossible differential for $2t-2$ rounds, as depicted below:

$$(0, \ldots, 0, \alpha) \xrightarrow[\text{prob. } 1]{R^{t-1}(\cdot)} (\alpha, 0, \ldots, 0) \neq (0, \ldots, 0, \beta) \xleftarrow[\text{prob. } 1]{R^{t-1}(\cdot)} (\beta, 0, \ldots, 0)$$

for $\alpha, \beta \neq 0$. Conservatively, $2t$ rounds will be secure against meet-in-the-middle attacks/distinguishers for the case $\kappa = n$, while $(2t-2)+(t+1) = 3t-1$ rounds will be secure against meet-in-the-middle attacks/distinguishers for the case $\kappa = t \cdot n$.

GMiMC$_{\text{Nyb}}$. There exists a probability-one truncated differential for a maximum of $t-1$ rounds of this construction (with $t = 2t'$ branches). This is described as follows:

$$(0, \alpha, 0, ..., 0) \to (\alpha, 0, 0, ..., 0) \to (*, 0, ..., 0, \alpha) \to (*, 0, ..., 0, \alpha, *) \to ... \to (*, 0, *, ..., *)$$

where $\alpha \neq 0$.

Using the probability-one truncated differentials similarly as described above, we can construct impossible differentials for GMiMC$_{\text{Nyb}}$. This will allow us to attack $2(t-1)$ rounds of the cipher. Hence, the number of iterations to protect the cipher against such attacks must be at least $2t$ for the case $\kappa = n$, and $2t+1$ for the case $\kappa = t \cdot n$.

11.4. Security Analysis – GMiMC instantiated over $\mathbb{F}_{2^n}$ in the Low-Data Attacks

For some practical applications considered in the following, we also consider the case in which the attacker has a limited access to data (e.g. 1 or 2 (plaintext, ciphertext) pairs). Here we consider this particular case, showing that, in some cases, the total number of rounds can be reduced (since many attacks cease to work in this particular scenario). Among all the attacks we have considered, only two of them apply for the case of low-data complexity, which are the GCD attack and its generalization as a Gröbner Basis attack. We emphasize that statistical attacks (like differential, linear, ...) are not competitive in this setting.

Gröbner Basis Analysis – Case: $\kappa = \mathbf{t} \cdot \mathbf{n}$. As explained in Sect. 11.3.1, the complexity of computing a Gröbner basis for a system of $\mathfrak{N}$ polynomials in $\mathfrak{V}$ variables is of $\binom{D_{reg}+\mathfrak{V}}{D_{reg}}$ operations over the base field $\mathbb{F}$ [BFP12], where D_{reg} is the degree of regularity. As already pointed out, closed-form formulas for D_{reg} are known only for some special cases (e.g. when $\mathfrak{N} = \mathfrak{V}$), but not in general.

In the low-data scenario, we use the SageMath code[19] in [AGP+18, App. H] to estimate D_{reg}, and so the complexity of the Gröbner Basis attack on GMiMC. In the low-data case, the analysis concludes that such attack do not outperform – in general – the GCD attack that we are going to present.

Greatest Common Divisors

Since the GCD attack is one of the few attacks that work in the low-data scenario, here we recall the idea of such an attack.

Given more than one known (plaintext, ciphertext) pair or working on the output of each branches of a single known (plaintext, ciphertext) pair (as described below), one can construct their polynomial representation. The idea of the GCD attack is simply to compute their polynomial Greatest Common Divisors (GCD) to recover a multiple of the key.

Two-pair case. Denote by $E(k, x)$ the encryption of x under key k. For a pair $(x, y) \in \mathbb{F}_{2^N}$, $E(K, x) - y$ denotes a univariate polynomial in $\mathbb{F}_q[K]$ corresponding to (x, y). Note that in our case the polynomial $E(K, x) - y$ can be constructed conceptually easily from the encryption process, but writing down $E(K, x) - y$ becomes computationally expensive as the number of rounds increases. Indeed, writing down $E(K, x) - y$ requires not only large computational resources but also an exponential (in r) amount of memory.

Consider now two such polynomials $E(K, p^1) - c^1$ and $E(K, p^2) - c^2$, with $c^i = E(k, p^i)$ for $i = 1, 2$ and for a fixed but unknown key k. It is clear that these polynomials share $(K - k)$ as a factor. Indeed, with high probability the greatest common divisor will be $(K - k)$. Thus, by computing the GCD of the two polynomials, we can find the value of k.

One-pair case. Since we are working with a Feistel construction, we can also set up a GCD computation among the branches of the Feistel cipher. In other words, let $p := (p_{t-1}, \ldots, p_1, p_0)$ and $c := (c_{t-1}, \ldots, c_1, c_0)$. For each component $i = 0, \ldots, t-1$, it is possible to construct the interpolation polynomial

$$c_i = E_i(K, (p_{t-1}, \ldots, p_1, p_0)),$$

where K is the secret variable. The analysis then proceeds as above, working on different components instead of different texts. Thus, it is possible to perform the GCD among the branches also in the case in which the attacker knows only 1 (plaintext, ciphertext) pair.

Meet-in-the-Middle. Due to the Feistel structure, a Meet-in-the-Middle variant of the GCD attack can be performed. That is, instead of constructing polynomials expressing ciphertexts as polynomials in the plaintext and the key, we can construct two polynomials $G'(K, x_i)$ and $G''(K, y_i)$ expressing the state in round $r/2$ as a polynomial in the key and the plaintext or ciphertext respectively. Then, considering $G'(K, x_0) - G''(K, y_0)$ and $G'(K, x_1) - G''(K, y_1)$, we can apply a GCD attack on polynomials with lower degree than before (approximately half).Hence, the number of rounds must be double to thwart this variant of the attack.

[19] ***Remark.*** *Since I did not work on such SageMath code – it was done by Martin R. Albrecht, I limit myself to refer to [AGP+18, App. H] for all details.*

Complexity. It is well-known that the complexity for finding the GCD of two polynomials of degree d is $\mathcal{O}(M(d) \log_2 d)$, where $M(d)$ is the cost of multiplying two degree-d polynomials. The best (known) complexity for $M(d)$ is $\mathcal{O}(d \log_2 d)$ using an FFT. Thus, we expect a GCD computation to cost $\mathcal{O}\left(d \log_2^2 d\right)$, where the hidden constant is greater than 1. In order to estimate the computational cost of such an attack, we have to estimate the degree of K in $E(K, x) - y$, which depends on the number of rounds r. To derive an estimate for the required number of rounds, we will target

$$d \log_2^2 d \approx 2^\kappa = 2^n,$$

where 2^κ denotes the computational cost of a brute-force attack and $\kappa = n$ denotes the number of key bits.

GMiMC$_{\mathbf{crf}}$. *Case: $\kappa = n$.* A key datum is the degree reached in each of our constructions after r rounds. Consider the t-branch, univariate case for GMiMC$_{crf}$ and denote the branches by $(X_{t-1}, \ldots, X_2, X_1, X_0)$. Given a plaintext, the degree d_i of the key in the i-th branch for $i = 0, \ldots, t-1$ after r rounds is

$$d_i = \begin{cases} 3^{r-i} & \text{if } r > i, \\ 0 & \text{otherwise.} \end{cases}$$

Note that $d_{t-1} = \min_i d_i$. The condition $3^{r-t+1} \log^2(3^{r-t+1}) \approx 2^n$ is fulfilled[20] when $r \simeq t - 1 + n \cdot \log_3 2 - 2 \log_3 n$. Thus, the number of rounds must be approximately

$$r \geq \lceil 2t + 2n \cdot \log_3 2 \rceil - \lfloor 4 \cdot \log_3 n \rfloor$$

to thwart the Meet-in-the-Middle variant.

Multivariate Case: $\kappa = t \cdot n$. To extend these attacks to the multivariate case, i.e. $\kappa = t \cdot n$, the attacker may guess $(t-1) \cdot n$ bits of the key, and then perform the previous GCD attack on a univariate polynomial. We note, however, that in the multivariate case we are targeting a complexity of $2^{t\,n}$ operations and are performing $2^{(t-1)\,n}$ GCD computations. Thus, each GCD computation has a “budget” of 2^n operations. On the other hand, guessing permits to shave off up to $(t-1)$ rounds. Thus, the number of rounds required in the multivariate case is slightly higher than in the univariate case. Of course, this trade-off changes when $2^{tn} \gg 2^\lambda$, where λ is the targeted security level.

As a result, the number of rounds must be approximately

$$r \geq \lceil 3t + 2n \cdot \log_3 2 - 1 \rceil - \lfloor 4 \cdot \log_3(n) \rfloor$$

to thwart the Meet-in-the-Middle variant. For each GMiMC family of block ciphers, we refer to Table 11.1 for the minimum number of rounds that ensure security against the GCD attack.

GMiMC$_{\mathbf{erf}}$. *Case: $\kappa = n$.* The degree d_i of the key in the i-th branch for $i = 0, \ldots, t-1$ after r rounds is

$$d_i = \begin{cases} 3^r & \text{if } r > 0 \text{ and } i \neq t-1, \\ 3^{r-1} & \text{if } r > 1 \text{ and } i = t-1, \\ 0 & \text{otherwise.} \end{cases}$$

Note that $d_{t-1} = \min_i d_i$. The condition $3^{r-1} \log^2(3^{r-1}) \approx 2^n$ is fulfilled when $r \simeq 1 + n \cdot \log_3 2 - 2 \log_3(n)$.

Finally, in order to prevent a modified version of this attack similar to the one proposed for the Gröbner basis case (see before), we increment this number of rounds by a factor $t - 3$. As a result, the number of rounds must be approximately

$$r \geq \lceil 2n \cdot \log_3 2 + 2t - 2 \rceil - \lfloor 4 \cdot \log_3(n) \rfloor$$

[20] Note that the solution of $y = x \cdot \log^2(x)$ is well approximated by $x = y / \log^2(y)$.

to thwart the Meet-in-the-Middle variant[21].

Case: $\kappa = t \cdot n$. The idea, once again, is simply to guess the first $t-1$ round keys (i.e., $(t-1)\cdot n$ bits of the key) and to apply the (univariate) GCD attack described previously. Using the previous strategy, it turns out that the number of rounds must be approximately

$$r = \lceil 2n \cdot \log_3 2 - 4 \cdot \log_3(n) + 3t - 3 \rceil$$

to thwart the Meet-in-the-Middle variant.

GMiMC$_{\mathbf{Nyb}}$. *Case:* $\kappa = n$. Consider the t-branch case with $t = 2 \cdot t'$. Since we are working in the univariate case, all the functions F_i are equal, i.e., $F_1 = F_2 = \cdots = F_t$, and they all depend on the same key. The degree d_i of X_i for $i = 0, \dots, t-1$ after $r \geq t'+1$ rounds[22] is

$$d_i = 3^{r-i+t'-1}.$$

In more detail, the degrees are given by

$$(3^{r-t'}, 3^{r-t'+1}, 3^{r-t'+2}, \dots, 3^{r-2}, 3^{r-1}, 3^{r}, 3^{r-1}, 3^{r-2}, \dots, 3^{r-t'+2}, 3^{r-t'+1}).$$

Note that $d_{t-1} = \min_i d_i$. The condition $3^{r-t'} \log_2(3^{r-t'}) \approx 2^n$ is fulfilled when $r \simeq t' + n \cdot \log_3 2 - 2\log_3(n)$. Thus, the number of rounds must be approximately

$$r \geq \lceil 2n \cdot \log_3 2 + t + 2 \rceil - \rfloor 4 \cdot \log_3(n) \lfloor$$

to thwart the Meet-in-the-Middle variant.

Case: $\kappa = t \cdot n$. Using the previous strategy and guessing the first $t-1$ round keys (which corresponds to skipping one round), it turns out that the number of rounds must be approximately

$$r = \lceil 2n \cdot \log_3 2 + t + 3 \rceil - \rfloor 4 \cdot \log_3(n) \lfloor$$

to thwart the Meet-in-the-Middle variant.

11.5. Parameter-Space Exploration

We compare the effects of different parameters in our Feistel-based constructions with block size N. In Table 11.3 we compare several parameters of the generalized constructions. Depending on the construction, a Feistel network may permit to compute more than one F function in parallel and we will refer to making use of this fact as "parallel mode". For example, in Nyberg's GFN mode $t/2$ (for $t \geq 4$) functions (in our case, multiplications) can be computed in parallel per round.

In the following, we propose some initial considerations. To simplify the notations, we denote the number of rounds necessary to protect the cipher from Interpolation attack, Gröbner basis attack, Higher-Order differential attack and (Truncated) Differential attack respectively by $R_{\text{Int}}, R_{\text{Gröbner}}, R_{\text{HighOrd}}, R_{\text{TDiff}}$.

Finally, in the following we denote by α the number of multiplication that must be performed to compute x^3 for an arbitrary x, i.e. $\alpha = 1$ for $x \in \mathbb{F}_{2^n}$ (where x^2 is linear in $\mathbb{F}_{2^n}$) and $\alpha = 2$ for $x \in \mathbb{F}_p$ (for prime p).

[21] We note that this attack crucially depends on separating monomials per round. In particular, if the degree of the target polynomial $\gg 2^n$, then this condition does not hold as modular reductions modulo $x^{2^n} - 1$ happen.

[22] We restrict to this case as we never consider $r < t'+1$ in our constructions.

Table 11.3.: Comparing the parameters of the GMiMC keyed permutation in different modes - *no restriction on data complexity.* To simplify the notations, we denote the number of rounds necessary to protect the cipher from Interpolation attack, Gröbner basis attack, Higher-Order differential attack and Truncated Differential attack respectively by $R_{\text{Int}}, R_{\text{Gröbner}}, R_{\text{HighOrd}}, R_{\text{TruncDiff}}$. Moreover, we use the notation n and N to denote respectively the cases $\log_2 p$ and $t \cdot \log_2 p$. Finally, α denotes the number of multiplication that must be performed to compute x^3 for an arbitrary x, i.e. $\alpha = 1$ for generic $x \in \mathbb{F}_{2^n}$ (where x^2 is linear in $\mathbb{F}_{2^n}$) and $\alpha = 2$ for generic $x \in \mathbb{F}_p$ (for prime p).

	Branches	**Security**	**round** (R)	**#mult**	**#mult** $\cdot\lvert\mathbb{F}\rvert$	**#mult**
		(κ bits)				**(parallel mode)**
MiMC	1	$n \equiv N$	$\log_3(2) \cdot n + 1$	$\alpha \cdot R$	$N \cdot \alpha \cdot R$	$\alpha \cdot R$
Feistel MiMC	2	n	$2 \cdot \log_3(2) \cdot n + 1$	$\alpha \cdot R$	$\frac{N}{2} \cdot \alpha \cdot R$	$\alpha \cdot R$
		N	$\lceil 2 \cdot \log_3(2) \cdot n \rceil + 3$	$\alpha \cdot R$	$\frac{N}{2} \cdot \alpha \cdot R$	$\alpha \cdot R$
GMiMC$_{crf}$	$t \geq 3$	n	$\max\{R_{\text{Int}}, R_{\text{HighOrd}}, R_{\text{TruncDiff}}\}$	$\alpha \cdot R$	$\frac{N}{t} \cdot \alpha \cdot R$	$\alpha \cdot R$
		N	$\max\{R_{\text{Int}}, R_{\text{Gröbner}}, R_{\text{HighOrd}}, R_{\text{TruncDiff}}\}$			
GMiMC$_{erf}$	$t \geq 3$	n	$\max\{R_{\text{Int}}, R_{\text{HighOrd}}, R_{\text{TruncDiff}}\}$	$\alpha \cdot R$	$\frac{N}{t} \cdot \alpha \cdot R$	$\alpha \cdot R$
		N	$\max\{R_{\text{Int}}, R_{\text{Gröbner}}, R_{\text{HighOrd}}, R_{\text{TruncDiff}}\}$			
GMiMC$_{Nyb}$	$t = 2t' \geq 4$	n	$\max\{R_{\text{Int}}, R_{\text{HighOrd}}, R_{\text{TruncDiff}}\}$	$\frac{t}{2} \cdot \alpha \cdot R$	$\frac{N}{2} \cdot \alpha \cdot R$	$\alpha \cdot R$
		N	$\max\{R_{\text{Int}}, R_{\text{Gröbner}}, R_{\text{HighOrd}}, R_{\text{TruncDiff}}\}$			

GMiMC$_{\text{crf}}$ vs GMiMC$_{\text{erf}}$. GMiMC$_{\text{crf}}$ and GMiMC$_{\text{erf}}$ are quite similar — only one multiplication is performed at each round. By our analysis, it turns out that GMiMC$_{\text{erf}}$ is always more efficient than GMiMC$_{\text{crf}}$, since it always requires a lower number of rounds to be secure. For this reason, we only consider GMiMC$_{\text{erf}}$ for the following practical applications.

Remark. As pointed out in the introduction, Feistel MiMC requires approximately double the number of rounds of MiMC. However, we found that the number of rounds does not grow linearly with the number of branches. For a concrete example, the cases of GMiMC$_{\text{erf}}$ with $t \cdot \log_2 p \approx 256$ and $t \cdot \log_2 p \approx 1024$ fixed are depicted in Fig. 11.4 and Fig. 11.5. It is possible to observe the minimum number of rounds is obtained by choosing the number of branches t not too "small" and not too "big" (e.g. $6 \leq t \leq 18$). As a result, for this range of values of t, GMiMC$_{\text{erf}}$ results to be as competitive as MiMC or even more for the applications that we have in mind. Similar results can be obtained for other values of $t \cdot \log_2 p$ and for all other GMiMC ciphers proposed here (we focus on GMiMC$_{\text{erf}}$ since it results to be the most competitive one for the practical applications that we are studying in this paper).

11.5.1. MPC/SNARK/PQ Signature Applications

In this section, we are interested to optimize the two GMiMC ciphers previously selected with respect to different metrics:

SNARK: minimize total number of "operations" – case $\kappa = n$ (we recall that SNARK applications use the hash function GMiMCHash, where the number of rounds of the inner permutation is given by the univariate case);

PQ Signature: minimize total number of multiplications × field size – case $\kappa = N$ in low-data scenario;

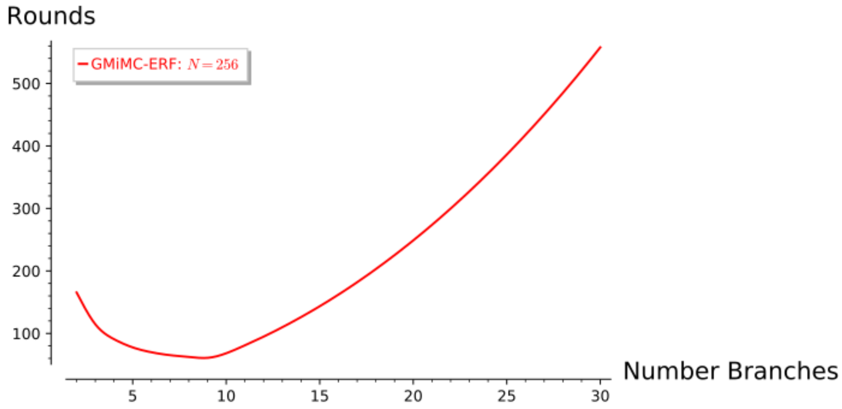


Figure 11.4.: Number of Branches *versus* Number of Rounds – $\text{GMiMC}_{\text{erf}}$ with $t \cdot \log_2 p \approx 256$ fixed.

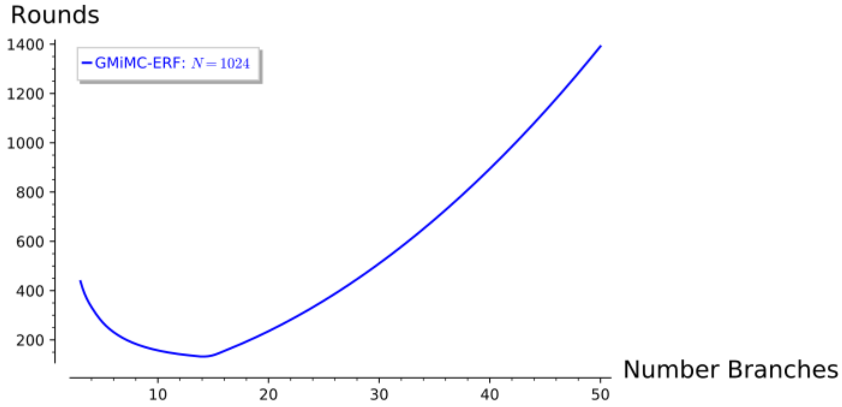


Figure 11.5.: Number of Branches *versus* Number of Rounds – $\text{GMiMC}_{\text{erf}}$ with $t \cdot \log_2 p \approx 1024$ fixed.

MPC: motivated by real life applications, our goal is to reduce the total runtime. The main bottleneck of a protocol ran on top of SPDZ-framework is the triple generation mechanism which is given by the number of (parallel) multiplications. Hence the goal is to minimize/optimize both the total number of operations (as for SNARKs) and the total number of (parallel) multiplications (where note that the two metrics coincide for GMiMC_{crf} and GMiMC_{erf}).

Remark. We remark that computing x^3 requires 2 multiplications in $\mathbb{F}_p$ and a single multiplication in $\mathbb{F}_{2^n}$ (since x^2 is linear in $\mathbb{F}_{2^n}$).

Focusing on $\mathbb{F}_{2^n}$, the cost of performing one multiplication in $\mathbb{F}_{2^n}$ using a fast Fourier transform is approximately $\mathcal{O}(n \cdot \log n)$ bit-wise XORs (that is, approximately $\beta \cdot n \cdot \log n$ for some constant β), while the cost of one addition is n bit-wise XORs. As a result, a good approximation of this number is given by

$$\text{number of rounds} \times \left(\mathcal{A} + \frac{N}{\beta \cdot n \cdot \log(n)}\right), \tag{11.4}$$

where

$$\text{GMiMC}_{crf}, \text{GMiMC}_{erf} : \mathcal{A} = 1 \qquad \text{and} \qquad \text{GMiMC}_{Nyb} : \mathcal{A} = t/2$$

since for each round $\mathcal{A}$ multiplication(s) and (approximately) $n \cdot t$ bit-wise XOR-sums are performed - remember that *(1st)* a single multiplication is necessary to compute x^3 in $\mathbb{F}_{2^n}$ and *(2nd)* the ratio between the cost of 1 multiplication and 1 addition is $1/\log(n)$. It follows that when the total number

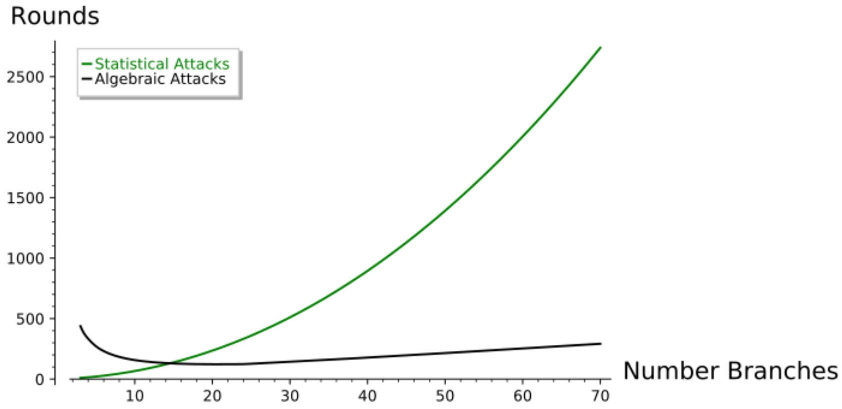


Figure 11.6.: Number of Branches *versus* Number of Rounds *of different attacks* – GMiMC$_{\text{erf}}$ with $t \cdot \log_2 p \approx 1024$ fixed. “Statistical Attacks” include Truncated diff., Impossible Diff. and Linear Diff. “Algebraic Attacks” include GCD attack, Interpolation attack, Gröbner Basis and Higher-Order Diff.

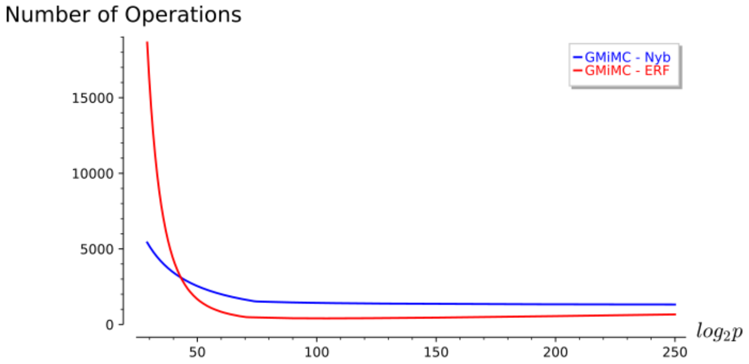


Figure 11.7.: SNARK in $\mathbb{F}_p$ - Comparison of the number of operations for GMiMC$_{erf}$ and GMiMC$_{Nyb}$ for different values of n ($N = 1024$).

of multiplications is higher than the total number of additions (e.g. MiMC or GMiMC$_{Nyb}$), it is reasonable to approximate the total number of operations by the total number of multiplications. When the total number of additions is much higher than the total number of multiplications, one must take care of both these two numbers to compute the total cost. Finally, in the PQ signature case, we primarily consider the total number of multiplications (and not of generic operations), since this metric determines both the signature size and the number of pseudo-randomly generated field elements required for signing.

Similar results can be obtained as well also for the case $\mathbb{F}_p$. Here the cost of 1 multiplication can be approximated as $O(\log_b(p)^2)$ word sum-operations (that is, approximately $\beta \cdot \log_b(p)^2$ for some constant[23] β) where b is the word size of the processor. Thus a good approximation for the total number of operations for the case $\mathbb{F}_p$ is given by

$$\text{number of rounds} \times \left(2 \cdot \mathcal{A} + \frac{t}{\beta \cdot [\log_2(p)]^2}\right), \tag{11.5}$$

where $p \approx 2^n$, $\mathcal{A}$ is defined as before, and the factor 2 counts for the two multiplications required to compute cubes in $\mathbb{F}_p$ (instead of a single one).

[23] For our practical implementation of the the SNARKs application in Section 11.6.2, the value of β is well approximated by $\beta \approx 1.75/32^2$ based on the multiplication and reduction algorithms used by NTL. This number is consistent with the complexity discussion of Karatsuba multiplication and Barrett reduction.

SNARK — Number of "Operations"

First of all, *for SNARKs applications we only consider the case* $\mathbb{F}_p$. This is motivated by the fact that we cannot use the property that squaring is linear in $\mathbb{F}_{2^n}$ for a more efficient implementation in this setting, since the cubes have to be represented as rank-1 constraint (see Section 11.6.2 for more details). We remark that similar consideration has been made for MiMC when used for SNARK applications (see [AGR+16, Section 6.1]).

Having said that and focusing only on the case $\kappa = n$, for the follow-up it is interesting to observe that for each N fixed, it is possible to minimize the total number of "operations" by adjusting the parameters t and n. Both for the case of GMiMC$_{Nyb}$ and GMiMC$_{erf}$, this number corresponds to the total number of multiplications. In GMiMC$_{erf}$, a good approximation of this number is given by formula (11.5).

Focusing on the case $N = 1024$ - used in the following application, it turns out that GMiMC$_{erf}$ is more efficient in this setting. Consider e.g. the case $\mathbb{F}_p$. As showed in Fig. 11.7, it turns out that the best choice for GMiMC$_{erf}$ is $n = 103$ and $t = 10$. The number of total operations for this choice of parameters is ≈ 403. In GMiMC$_{Nyb}$, the number of multiplications is given by $R \times t$ (where R is the number of rounds and $2 \times (t/2) = t$ multiplications in $\mathbb{F}_p$ are performed in each round), which means that the total number of operations is (at least)

$$\max\left\{1.262 \cdot N + 3t + t^2, 0.631 \cdot N + 2t \cdot \log_3(t) + t^2 + t, 3t + 2t^2 + 4t \cdot \log_2(N)\right\} \geq 1.262 \cdot N \approx 1293,$$

that is approximately three times more. As a result, for the case $N = 1024$, GMiMC$_{erf}$ is (much) more efficient than GMiMC$_{Nyb}$.

PQ Signatures — Number of Multiplications × Field Size (Low-Data Scenario)

Focusing only on the case $\kappa = N = t \cdot n$, for the follow-up it is interesting to observe that for N fixed, it is possible to minimize the product of multiplications and branch size by adjusting the parameters t and n. This metric is the most interesting one, since it determines both the signature size and the size of the random tapes. Remember that the PQ signature is implemented in the low-data scenario only (which means that e.g. the differential attack does not apply). We give the best choice of t and n for this metric:

GMiMC$_{\text{erf}}$ With respect to the general scenario, in this case there is no closed-form formula to compute the number of rounds necessary to guarantee security. Combining the results provided by the GCD attack and the one provided by the SageMath code given in [AGP+18, App. H] (in order to estimate Gröbner Basis attack), the number of rounds to provide security is $r \geq \lceil 1.262 \cdot n - 4 \cdot \log_3(n) \rceil + 3t + 3$ and the best choice is

$$n = 3, \qquad r \cdot n \geq \lceil 1.262 \cdot n^2 - 4 \cdot \log_3(n) \cdot n \rceil + 3N + 9 \geq 3N + 9.$$

GMiMC$_{\text{Nyb}}$ Since the number of multiplications is given by $r \cdot t/2$, it follows that[24]

$$r \cdot \frac{t}{2} \cdot n \geq \lceil 0.631 \cdot N \cdot n \rceil + N \cdot t + \frac{N}{2},$$

which is higher than the corresponding number for GMiMC$_{erf}$. Indeed, since the best choice is to minimize n also in this case (that is, to choose $n = 3$), it follows that $r \cdot \frac{t}{2} \cdot n \geq N \cdot (\frac{N}{2} - 0.6) = \mathcal{O}(N^2)$ (*vs* $\mathcal{O}(N)$ for the case of GMiMC$_{erf}$).

In more detail, for $n = 3$: $\underbrace{N^2/3 + 2.4N}_{\text{GMiMC}_{Nyb}} \geq \underbrace{3N + 9}_{\text{GMiMC}_{\text{erf}}}$ for each $N \geq 7$.

As a result, it follows that GMiMC$_{erf}$ is more efficient in this setting (analogous for $\mathbb{F}_p$).

[24] We use the number of rounds provided by the GCD attack for the given estimation. Note that the real number of rounds is not lower than the number of rounds of the GCD attack.

Mode	**#Branch /**	**Online cost**				**Preproc**
	#Block	**(MPC) Rounds**	**Openings**	**Latency** (ms)/$\mathbb{F}_p$	**Throughput** $\mathbb{F}_p/s$	(ms)
GMiMC$_{crf}$	8	386	579	3.70	34310	12.8
GMiMC$_{erf}$		356	534	3.41	37813	11.8
GMiMC$_{Nyb}$		344	2064	3.27	11329	45.8
GMiMC$_{mrf}$		344	2064	3.36	11591	45.8
MiMC	8 blocks	146	1752	1.39	13846	38.9
GMiMC$_{crf}$	64	834	1251	1.00	75967	27.8
GMiMC$_{erf}$		580	870	0.68	111047	19.3
GMiMC$_{Nyb}$		456	21888	0.63	9380	486
GMiMC$_{mrf}$		356	17088	0.49	12153	379
MiMC	64 blocks	146	14016	0.21	14814	311

Table 11.4.: Two-party costs for MiMC and GMiMC over a LAN network.

MPC — Number of (parallel) Multiplications

In MPC the number of communications rounds is equal to the number of (parallel) multiplications - that is $\alpha \cdot R$ where α and R are defined as before - for all the proposed designs. In particular, note that for GMiMC$_{Nyb}$ and GMiMC$_{mrf}$ the $t/2$ multiplications can all be executed in parallel. On the other hand, these parallel multiplications are not "costless": the effect of these $t/2$ multiplications per round is reflected in the throughput metric.

When N is fixed, GMiMC$_{Nyb}$ requires a lower number of rounds (for both encryption and MPC communications) than GMiMC$_{erf}$ to achieve security. However, GMiMC$_{mrf}$ has significantly less throughput compared to GMiMC$_{erf}$.

11.6. Applications

Remark. *Since I did not work on the practical applications/implementations of GMiMC, I limit myself to recall here the main results and I refer to [AGP+18] for a detailed discussion on it. The results of this section are due to the work of Dragos Rotaru (MPC applications), Arnab Roy (SNARKs application) and Sebastian Ramacher and Markus Schofnegger (PQ-Signature application) respectively.*

11.6.1. MPC Applications

Benchmarking Environment. We have benchmarked the protocols using the SPDZ framework, which provides active security against multiple malicious parties [KSS13]. Additions of secret values and scalar multiplications are (almost) for free in SPDZ. The protocols ran across two computers with commodity hardware connected via a 1 GB/s LAN network and an average round-trip time of 0.3 ms. In our setting, both keys and messages are secretly shared among the two parties.

Results. For a complete measurement of an MPC protocol, one needs to have in mind both pre-processing and online phases. The pre-processing phase cost is determined by the number of shared multiplications. Performance of the online phase is given by the multiplicative depth of the circuit to be evaluated as well as the number of openings (whenever a party reveals a secret value). For the online phase we give measurements in terms of *latency* and *throughput*. Latency indicates the time spent for computing a single block cipher call, whereas throughput shows the maximum $\mathbb{F}_p$ objects that can be encrypted in parallel per second. We instantiate each block cipher with 8 and 64 input blocks/branches, where each block lies in $\mathbb{F}_p$ and $p \approx 2^{128}$. Note that for GMiMC constructions in MPC we have used an n-bit key. For a fair comparison with previous evaluations of

Mode	#Branch /	Online cost				Preproc
	#Block	(MPC) Rounds	Openings	Latency (ms)/$\mathbb{F}_p$	Throughput $\mathbb{F}_p/s$	(ms)
GMiMC$_{crf}$	8	386	579	2427	1937	120.6
GMiMC$_{erf}$		356	534	2249	1500	111.2
GMiMC$_{Nyb}$		344	2064	2173	401	430
GMiMC$_{mrf}$		344	2064	2149	401	430
MiMC	8 blocks	146	1752	7421	644	365
GMiMC$_{crf}$	64	834	1251	659	6945	260.6
GMiMC$_{erf}$		580	870	459	7303	181.2
GMiMC$_{Nyb}$		456	21888	361	354	4560
GMiMC$_{mrf}$		356	17088	280	471	3560
MiMC	64 blocks	146	14016	116	646	2920

Table 11.5.: Two-party costs for MiMC and GMiMC over a WAN network.

MiMC in SPDZ, the online phase runs on a single thread. The preprocessing column denotes the amount of time required to generate the triples for a single block cipher evaluation in a two party SPDZ protocol.

Experiments (Table 11.4) show that GMiMC$_{crf}$ and GMiMC$_{erf}$ have a very fast pre-processing phase because they perform a low number of multiplications. A big advantage of these two is how well they scale in terms of triples used, since they require one multiplication per round. This is in contrast with MiMC, where increasing the parallelism by a factor of c results in c times more multiplications per round.

Perhaps unexpectedly, GMiMC$_{crf}$ and GMiMC$_{erf}$ have a higher throughput compared to the rest of the variants, although they result in a larger number of rounds. The reason is that fewer openings - or multiplications in our case - mean less data sent between the parties so we can batch more executions in parallel. Thus in a LAN network the number of rounds has a minor impact.

The situation looks similar in a WAN network (Table 11.5). Even higher throughput improvements, yet we notice that a higher number of rounds affects the latency more significantly than in the LAN case. We can see that GMiMC$_{crf}$ and GMiMC$_{erf}$ win against the classic MiMC variant by at least one order of magnitude in terms of throughput and pre-processing material.

11.6.2. SNARKs Applications

Benchmarking Environment. For all the field operations we have used the NTL together with the `gf2x` library. All the computations were performed on a system having an Intel Core i7-6700 @3.4 GHz ×8 processor with 16 GB memory. We took the average time over ≈ 2000 iterations.

Results. Since we expect that GMiMC$_{crf}$ and GMiMC$_{erf}$ work better than GMiMC$_{Nyb}$ in a SNARK setting, we limited to implement them for the GMiMC permutation and hash function. We compared the performance with MiMC. For $N \approx 1024$-bit (prime) block size GMiMC$_{erf}[N, t, n]$, where $t = 8$, shows some improvement over MiMC-1025. For hashing a single message block, GMiMCHash-256 is more than 1.2 times faster than MiMCHash-256 and is significantly (> 12 times) faster than SHA-256. We stress that in comparison with MiMCHash the primary advantage of GMiMC$_{erf}$Hash is that it can be used over 256 bit or smaller field size.

Note that the number of constraints for GMiMCHash-256 with fixed key permutation is only one more than the number of constraints for $GMiMC_{\text{erf}}$. Hence the time taken by the hash function and the permutation with fixed key are the same (in Table 11.6).

	MiMC [AGR+16]		GMiMC$_{\mathbf{erf}}$		
$(t, \log_2(p), R)$	$(1, 1024, 646)$	$(2, 513, 647)$	$(4, 256, 332)$	$(8, 128, 178)$	$(16, 64, 141)$
constraint generation	4.553 ms	5.077 ms	4.735 ms	4.732 ms	8.057 ms
witness generation	1.079 ms	0.639 ms	0.388 ms	0.296 ms	0.449 ms
total time	5.632 ms	5.716 ms	5.123 ms	5.028 ms	8.507 ms
#additions	646	1293	996	1246	2115
#multiplications	1293	1293	664	356	282

Table 11.6.: Comparison of MiMC with GMiMC$_{erf}$ (with different numbers of branches) in SNARK in $\mathbb{F}_p$ when the block size is 1024 bits.

11.6.3. Post-Quantum Signature Applications

Fish and Picnic [CDG+17] are new classes of digital signature schemes which derive their security entirely from the security of symmetric-key primitives, have extremely small key pairs, and are highly parameterizable. The construction is based on a one-way function f, where for the secret key x, the image $y = f(x)$ is published as the public key. A signature on a message is then obtained from a non-interactive zero-knowledge proof of the relation $y = f(x)$, that incorporates the message in the challenge generation. When instantiating f with LowMC [ARS+15], trying to reduce the signature size by reducing the number of multiplication gates comes at the cost of a more expensive linear layer, which leads to a runtime vs. signature-size trade-off. Since the security proofs in [CDG+17] only require a block cipher with a reduced data complexity of 1, the overall performance can be greatly improved as this fact allows to choose LowMC instances with less rounds. For the 128 bit PQ security level, e.g. 256-bit block size and key size, a good trade-off can be found by using 10 S-Boxes and 38 rounds, resulting in a view size of 1140 bits.

Benchmarking Environment. All the computations were performed on a system having an Intel Core i7-4790 with 3.6 GHz.

Results. We implemented the signature scheme using GMiMC$_{erf}$ with key size and block size of ≈ 256 bits to build the one-way function, considering only the low-data scenario. In Table 11.7 we compare MiMC and GMiMC$_{erf}$ with different numbers of branches. We also include the view size required per repetition of ZKB++ stored in the signature and the runtime of the encryption of a single block. As we increase the number of branches, the expected signature sizes decrease.

As a result, even for very small fields with the smallest signatures but slower signing and verification, GMiMC$_{erf}$ performs significantly better in terms of signature size and runtime than MiMC. We also note that the implementation over $\mathbb{F}_{2^n}$ generally performs better than the comparable parameterization in the prime field case. In the binary case we can either follow the same approach as in the prime case and implement the circuit using $\mathbb{F}_{2^n}$ arithmetic where the permutation requires two multiplications. When implementing the circuit using $\mathbb{F}_2$ arithmetic, we have to emulate $\mathbb{F}_{2^n}$ operations in the circuit, but we end up with smaller view sizes and a more efficient implementation. Compared to LowMC, choosing an instance over $\mathbb{F}_{2^3}$ allows us to beat the smallest signatures sizes obtainable using LowMC with one S-Box by 306 bit in terms of view size.

11.6.4. Conclusion

One key take-away of this work is that, when it comes to building structures in symmetric cryptography with low multiplicative complexity, balanced Feistel networks are not the best strategy. We

Scheme	$(\mathbf{n},\mathbf{t},\mathbf{R})$	**Sign**	**Verify**	**View Size**
MiMC [AGR+16]	$(256,1,162)$	333.97 ms	166.28 ms	83456 bits
	$(272,1,172)$	92.45 ms	46.32 ms	94112 bits
GMiMC$_{\text{erf}}$ over $\mathbb{F}_p$	$(3,86,261)$	97.32 ms	72.06 ms	1566 bits
	$(4,64,196)$	62.35 ms	45.16 ms	1568 bits
	$(16,16,62)$	7.59 ms	5.13 ms	1984 bits
	$(32,8,55)$	4.95 ms	3.05 ms	3520 bits
	$(64,4,81)$	11.78 ms	6.85 ms	10368 bits
	$(136,2,163)$	67.51 ms	35.21 ms	44336 bits
GMiMC$_{\text{erf}}$ over $\mathbb{F}_{2^n}$	$(3,86,261)$	16.06 ms	10.76 ms	**783 bits**
	$(17,16,63)$	*3.73 ms*	*2.30 ms*	*1071 bits*
	$(33,8,56)$	3.34 ms	2.29 ms	1848 bits
	$(65,4,82)$	6.47 ms	4.02 ms	10660 bits
LowMC [ARS+15]	$(256,10,38)$	3.74 ms	3.52 ms	1140 bits
	$(256,1,363)$	9.55 ms	7.12 ms	1089 bits

Table 11.7.: Comparison of MiMC with GMiMC$_{erf}$ and LowMC when the block size is ≈ 256 bits in the context of Fish. For LowMC, n corresponds to the block size, t is the number of S-Boxes, and R denotes the number of rounds. Runtimes given for *Sign* and *Verify* are for the circuit computations only.

provided a new and optimal (in some sense) variant of the GFN and yet we still cannot beat the ERF variant.

This observation is surprising (and thus interesting): Unbalanced Feistel networks, which appeared no later than the late 1980s, do not have a great track record in the academic literature and in recent designs. As an illustration, consider that among all the lightweight block cipher designs listed on the CryptoLux lightweight block cipher wiki[25], seven are Type-II GFNs and ten are balanced Feistel networks, whereas *none* is of the UFN or ERF type.

And yet exactly those types turn out to be the best in our setting. We can even make a parallel with MiMC itself: Its structure is strongly related and building up on a design from the mid 1990s, which in recent textbooks [KR11, Sect. 8.4] was even shown as an example of how not to design a cipher. Despite this fact, it has turned out to be very good in many applications where multiplicative complexity matters. It may well be that the same is true with our work: Cryptographers had lost interest in the UBF or never considered it a reasonable option and yet it is the best in several of our specific use cases.

[25] `https://www.cryptolux.org/index.php/Lightweight_Block_Ciphers`

12
Hades Strategy and HadesMiMC

Another possible generalization of MiMC is HadesMiMC, constructed using the Hades strategy [GLR+19]. Hades *strategy is a high-level design approach for cryptographic permutations and keyed permutations* addressing needs of new applications that emphasize the role of multiplications in such designs, with a focus on simple arguments for its security. *It builds up on the Wide-Trail design strategy for SP-Networks*, which proved already very useful for a plethora of cipher and permutation designs as it helps to argue security against important classes of cryptanalytic attacks such as differential or linear attacks in a clean and simple way. *Our approach "Hades" additionally allows for such arguments against important classes of algebraic attacks that are of much more concern when multiplications are to be minimized in a design.*

In order to set up a strategy which is simultaneously elegant, simple and that provides security arguments a larger number of classes of attacks, and at the same time results in the most competitive instantiations to date, we use a freedom in the design space that was so far not exploited: *moving from an even to a highly uneven distribution of non-linearity.*

For our concrete instantiation "HadesMiMC", we borrow ideas from the pre-predecessor of Rijndael/AES, namely SHARK [RDP+96], an S-Box based design with a single large MDS layer covering the whole internal state. Main applications of HadesMiMC are PQ digital signature scheme, and MPC evaluations. For MPC, compared to the currently fastest design MiMC, our current analysis suggest a significant improvements in throughput and simultaneously a reduction of preprocessing effort, albeit at the cost of a higher online latency. In the PQ-Signature use-case, we are currently able to achieve the fastest signing times and lowest signatures sizes possible to date, even outperforming LowMC.

12.1. Introduction and Motivations

Wide Trail Strategy. Many modern block ciphers and permutations are designed based on the concept of an *iterative block cipher/permutation*, that is, a construction that consists of the repeated composition of (simple) functions. One widespread implementation of such ciphers is the substitution-permutation network (SPN). It takes a block of the plaintext and the key as inputs, and applies several alternating "rounds" of non-linear substitution boxes (S-Boxes) and linear permutation boxes (P-boxes) to produce the ciphertext block.

The *wide trail strategy* [DR01; DR02a] is an approach to design round transformations of block ciphers that combine efficiency and resistance against differential and linear cryptanalysis, probably the most common and efficient techniques in cryptanalysis. Instead of spending most of its resources on large S-Boxes, the wide trail strategy aims at *designing the round transformation(s) in order to maximize the minimum number of active S-Boxes over multiple rounds.* Thus, in ciphers designed with the wide trail strategy, a relatively large amount of resources is spent in the linear step to provide high multiple-round diffusion. Designing block ciphers and hash functions in a manner that resemble the AES in many aspects has been very popular since Rijndael was adopted as the Advanced Encryption Standard (AES) [DR02b], currently the de facto block cipher standard, known for its elegant and simple design, high security, and efficiency.

Partial S-Box Layers. On the other hand, the wide trail strategy can clearly not guarantee security against all attacks in the literature. As a concrete example, algebraic attacks that exploit the low degree of the encryption or decryption function – like the interpolation attack [JK97] or the higher-order one [Knu94] – are (almost) independent of the linear layer used in the round transformation[1], which is the crucial point of such a design strategy. In other words, *especially in the case of a low-degree S-Box, the wide trail strategy is not sufficient by itself, and it must be combined with "something else" (e.g., increasing the number of rounds) in order to guarantee security against all possible attacks in the literature.*

Moreover, *the "hidden" assumption of such a strategy is that each round contains a full S-Box layer.* Even if this is a well accepted practice (linear and non-linear building blocks have roughly similar costs in hardware and software implementations), there are various applications/contexts in which the cost of a non-linear operation is much higher than the cost of a linear operation. As concrete examples, for *masking* and practical applications of secure multi-party computation (MPC), fully homomorphic encryption (FHE), and zero-knowledge proofs (ZK) that use symmetric primitives, the linear computations are often much cheaper than non-linear operations.

A possible way to achieve a lower implementation cost is by *specializing a block cipher by minimizing the number of non-linear operations.* To achieve this goal, possible strategies are looking for low-degree S-Boxes and/or exploiting SPN structures where not all the state goes through the S-Boxes in each round. This second approach has been proposed for the first time by Gérard *et al.* [GGNS13] at CHES 2013. Such partial non-linear SP networks – in which the non-linear operation is applied to only a part of the state in every round – contains a wide range of possible concrete schemes that were not considered so far, some of which have performance advantage on certain platforms. As a concrete instantiation of their methodology, Gérard *et al.* designed Zorro [GGNS13], a 128-bit lightweight AES-like cipher which reduces the application of the S-Box per round, from 16 to only 4.

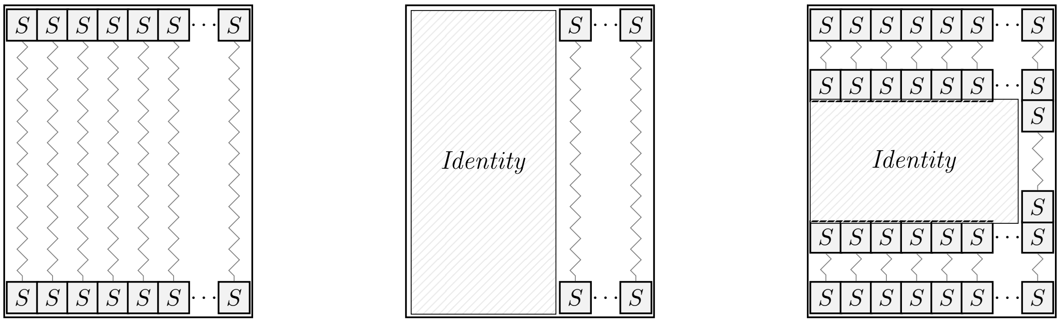
A similar approach has also been considered by Albrecht *et al.* [ARS+15] in the recent design of a family of block ciphers called LowMC proposed at Eurocrypt 2015. LowMC is a flexible block cipher (with very small multiplicative size and depth) based on an SPN structure and designed for MPC/FHE/ZK applications. Similar to Zorro, a partial non-linear layer is adapted in LowMC design. Basically LowMC combines an incomplete S-Box layer with a strong linear layer to reduce the multiplicative depth and size of the cipher.

How Risky are Partial SP-Networks? Due to their innovative designs, the wide trail strategy and the tools that were developed in order to formally prove the security of block ciphers against standard differential and linear cryptanalysis do not apply to Partial-SP-Networks such as Zorro and LowMC. For this reason, authors replaced the formal proof by heuristic arguments.

For the case of Zorro, the simple bounds on the number of active S-Boxes in linear and differential characteristics[2] cannot be used due to the modified SubBytes operation. Even though authors come up with a dedicated approach to show security of their design, such heuristic argument turned out to be insufficient, as Wang *et al.* [GNPW13; WWGY14] found iterative differential and linear characteristics that were missed by the heuristic and used them to break full Zorro. To fix this problem, an automated characteristic search tool and dedicated key-recovery algorithms for SP networks with partial non-linear layers have been presented at [BDD+15]. The generic techniques for differential and linear cryptanalysis of SP networks with partial non-linear layers proposed in there can be used both for cryptanalysis of such schemes and for proving their security with respect to basic differential and linear cryptanalysis, succeeding where previous automated analysis tools seem to fail. Beside obtaining practical attacks on Partial-SPN ciphers, authors concluded that even

[1] We remark that a linear/affine function does not increase/change the degree.

[2] We recall that 24-round Zorro, when compared to 10-round AES, has 40% less non-linear operations, but 140% more linear ones.



(a) SPN **(b)** P-SPN **(c)** Hades

Figure 12.1.: SP-Networks and Generalizations (P-SPNs and Hades).[3]

if "*the methodology of building PSP networks based on AES in a straightforward way is flawed, [...] the basic PSP network design methodology can potentially be reused in future secure designs*".

Similar, authors of LowMC chose the number of rounds in order to guarantee that no differential/linear characteristic can cover the whole cipher with *non-negligible probability.* However, they do not provide such strong security arguments against other attack vectors including algebraic attacks. As a result, the security of earlier versions of LowMC against algebraic attacks was found to be lower than expected [DLMW15; DEM15], and full key-recovery attacks on LowMC have been set up. For example, the incomplete S-Box layer also facilitates the existence of linear relations with probability 1, which allow to attack additional rounds. More recently, generalizations of impossible differential attacks have been found for some LowMC instances [RST18].

The Idea in a Nutshell - The Hades Strategy

Summarizing the current situation: Wide-trail strategy is appealing due to its simplicity, but limited to differential and linear attacks, and does not work with partial S-Box layers. Additionally, when S-Boxes are chosen to have low degree, not most relevant attack vectors are others anyhow. Designs of this type, like Zorro and LowMC, require a lot of ad-hoc analysis.

To address this issue we propose to start with a classical wide-trail design, i.e. with a full S-Box layer (outer layer), and then add a part with full and/or partial S-Box layers in the middle. Even without the middle part, the outer layer in itself is supposed to give arguments against differential and linear attacks in exactly the same way the wide-trail strategy does. At the same time, arguments against low-degree attacks can be obtained working on the middle layer. Since algebraic attacks mainly exploit the small degree of the encryption/decryption functions, the main role of this middle part is to achieve high algebraic degree, with perhaps only few (e.g. one) S-Boxes per round. On the other hand, the cost of algebraic attacks can also depend – in general – on other factors besides the degree of the function. Depending on the cost metric of the target application that one has in mind (e.g. minimizing total number of non-linear operations), we show that the best solution is to choose the optimal ratio between the number of rounds with full S-Box layer and with partial S-Box layer in order to achieve both security and performance.

We refer to this high-level approach as the "Hades Strategy"[4], and will be more concrete in the following.

Related Work – Designs with Different Round Functions. Almost all designs for block ciphers and permutations, not only those following the wide-trail design strategy, use round functions

[3] ***Acknowledgement.*** *Figure 12.1 – made by Markus Schofnegger – has been copied from [GLR+19].*

[4] *Why "Hades Strategy"?* Referring to Fig. 12.2, if one highlights the S-Boxes per round, the obtained picture resembles a "*bident*". In classical mythology, the bident is a weapon associated with Hades, the ruler of the underworld. The name of our strategy comes from here.

that are very similar, differing often only in so-called round constants which break symmetries in order to prevent attacks like slide attacks. Notable exceptions to this are the AES finalist MARS [IBM] and PRINCE [BCG+12]. MARS has whitening rounds with a different structure than the inner rounds with the idea to frustrate cryptanalytic attacks. A downside was perhaps that it also complicated cryptanalysis. PRINCE rounds differ in that the later half of the rounds is essentially the inverse of the first half of the rounds, and a special middle round is introduced. This allows to achieve a special property, namely that a circuit describing PRINCE computes its own inverse (when keyed in a particular way). Finally, we mention the cases of LowMC [ARS+15] and Rasta [DEG+18], for which different (independent and random) linear layers are used in each round. Due to their particular design strategies, this allows to maximize the amount of diffusion achieved by the linear layer.

In none of these cases, however, the *amount* on non-linearity, and hence their cryptographic strength, differs over the rounds.

12.2. Description of the Hades Strategy and HadesMiMC

12.2.1. Hades Strategy

Block ciphers are typically designed by iterating an efficiently implementable round function many times in the hope that the resulting composition behaves like a randomly drawn permutation. In general, *the same round function is iterated enough times to make sure that any symmetries and structural properties that might exist in the round function vanish.*

Instead of considering the same round function in order to construct the cipher (to be more precise, the same non-linear layer for all rounds), we propose - for the first time in the literature - to consider *a variable number of S-Boxes per round*, that is, to use different S-Box layers in the round functions.

Similar to any other SPN design, each round of a cipher based on Hades is composed of three steps:

1. *Add-Round Key* - denoted by $ARK(\cdot)$;
2. *SubWords* operation - denoted by S-Box$(\cdot)$;
3. *MixLayer* - denoted by $M(\cdot)$.

A final round key addition is then performed, and the final MixLayer operation can be omitted (we sometimes include it in this description for simplicity):

$$\underbrace{ARK \to \text{ S-Box} \to M}_{1st \text{ round}} \to ... \to \underbrace{ARK \to \text{ S-Box} \to M}_{(R-1)\text{-}th \text{ round}} \to \underbrace{ARK \to \text{ S-Box}}_{R\text{-}th \text{ round}} \to ARK$$

The crucial property of HadesMiMC is that *the number of S-Boxes per round is not the same for every round*:

- a certain number of rounds - denoted by R_F - has a *full* S-Box layer, i.e., t S-Box functions;
- a certain number of rounds - denoted by R_P - has a *partial* S-Box layer, i.e., $1 \leq s < t$ S-Boxes and $(t-s)$ identity functions.

In general, we limit ourselves to consider the case $s = 1$, that is, R_P rounds have a single S-Box per round and $t-1$ identity functions. However, we remark that this construction can be easily generalized (e.g. like LowMC) allowing more than a single S-Box per round in the middle R_P rounds.

In more details, assume $R_F = 2 \cdot R_f$ is an even number[5]. Then

[5] $R_F = 2 \cdot R_f$ is even in order to have a "symmetric" cipher. Note that some attacks – like the statistical ones – have the same performance both in the encryption and in the decryption direction. Thus a "symmetric" cipher with $R_F = 2 \cdot R_f$ guarantees the same security against these attacks both in the chosen-/known-plaintext scenario and in the chosen-/known-ciphertext one.

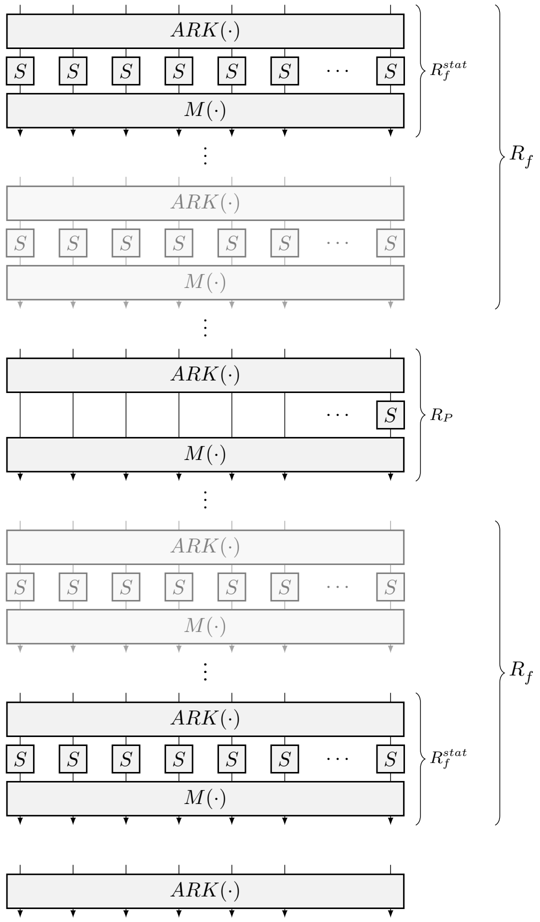


Figure 12.2.: Construction of HADES (the final matrix multiplication can be omitted).[6]

- the first R_f rounds have a full S-Box layer,
- the middle R_P rounds have a partial S-Box layer (i.e., 1 S-Box layer),
- the last R_f rounds have a full S-Box layer.

Figure 12.2 shows the strategy HADES. Note that the rounds with a partial S-Box layer are "masked" by the rounds with a full S-Box layer, which means that an attacker should not (directly) take advantage of the rounds with a partial S-Box layer.

Crucial Points of Hades Strategy. The crucial point of our design is that it contains *both rounds with full S-Box layers and rounds with partial S-Box layers*. This allows to provide *simpler argumentation about the security against statistical attacks* than the one proposed for P-SPN ciphers.

In more details, a certain number of rounds $R_F^{stat} = 2 \cdot R_f^{stat}$ with full S-Box layer situated at the beginning and the end guarantee security against statistical attacks. Indeed, even without the middle part, they are sufficient in order to apply the "Wide-Trail" strategy, in a way that we are going to show in the following. Security against all algebraic attacks is achieved working both with rounds $R_F = R_F^{stat} + R'_F \geq R_F^{stat}$ with full S-Box layers and rounds $R_P \geq 0$ with partial S-Box layers. Even if few (even one) S-Boxes per round are potentially sufficient to increase the degree of the encryption/decryption function (which mainly influences the cost of an algebraic attack), other factors can play a crucial role on the cost of such attacks (e.g. a Gröbner basis attack depends also on the number of non-linear equations to solve).

[6] ***Acknowledgement.*** *Figure 12.2 – made by Markus Schofnegger – has been copied from [GLR+19].*

With this in mind, the idea is to construct "something in the middle" between an SPN and a P-SPN cipher. Moreover, since we aim to have the same security w.r.t. chosen-plaintext and chosen-ciphertext attacks, we consider a cipher which is "symmetric": in other words, the same number of rounds with full non-linear layers are applied at the beginning and at the end, where the rounds with partial non-linear layers are in the middle and they are "masked" by the rounds with full non-linear layers. As a result, depending on the cost metric that one aims to minimize (e.g. the total number of non-linear operations) and on the size of the S-Box, in the following we provide the *best ratio* between the number of rounds with full S-Box layers and with partial ones in order to both achieve security and minimize the cost metric.

What about the choice of the linear and of the non-linear layer? Our strategy does not pose any restriction/constriction on the choice of the linear layer and/or on the choice of the S-Box. The idea is to *consider a "traditional" SPN cipher based on the wide trail strategy, and then to replace a certain number of rounds with full S-Box layer with the same number of rounds with partial S-Box layer* in order to minimize the number of non-linear operations, but without affecting the security. The Hades strategy has a huge impact especially in the case of ciphers with low-degree S-Box, since in this case a large number of rounds is required to guarantee security against algebraic attacks.

12.2.2. The Block Cipher HadesMiMC

HadesMiMC is a block cipher constructed using the strategy just proposed, hence it is both an SPN and a Partial-SPN cipher. Roughly speaking, HadesMiMC is obtained by applying the Hades strategy to the cipher Shark [RDP+96], proposed by Rijmen *et al.* in 1996 and based on the wide trail strategy.

HadesMiMC works with texts of $t \geq 2$ words[7] in $\mathbb{F}_p$ or $\mathbb{F}_{2^n}$, where p is a prime of size 2^n (in the following, $p \approx 2^n$). For simplicity, we limit ourselves to describe HadesMiMC for the $\mathbb{F}_{2^n}^t$ case[8], but the same description applies also to the $\mathbb{F}_p$ case. In the following, let $N := n \cdot t \simeq \log_2 p \cdot t$.

As for Shark, the MixLayer of HadesMiMC is simply defined by a multiplication with a fixed $t \times t$ near-MDS or MDS matrix (more generally, a matrix with an high branch number). The number of rounds $R = 2 \cdot R_f + R_P$ depends on the choice of the S-Box and of the parameters n and t. For the applications that we have in mind, we focus only on the cubic S-Box, S-Box$(x) = x^3$, where remember that S-Box$(x) = x^3$ is a permutation in $GF(2^n)$ iff n is odd and that it is a permutation in $GF(p)$ iff $p \neq 1 \mod 3$ (see "Hermite's criterion" for more details).

More details about the near-MDS/MDS matrix and the key-schedule are given in the following.

Why SHARK among Many Others? Since in our practical applications the cost of linear operations is much less (roughly speaking, "negligible") than the cost of non-linear ones, we decided to consider the most "efficient" linear layer in order to construct HadesMiMC. This corresponds to a linear layer defined as a multiplication with an MDS matrix that involves the entire state, which is exactly the case of SHARK.

Since our design strategy can be applied to any SPN design, a possible future problem is to apply Hades to e.g. AES, that is, to check if a certain number of rounds of AES can be replaced with rounds that contain only a partial non-linear layer (e.g. one or four S-Boxes) without decreasing its security.

Choice of the S-Box. Before going on, we mention that we also considered possible variants of HadesMiMC instantiated by S-Boxes defined by a different power exponent. Since our main goal

[7] The case $t = 1$ corresponds to MiMC [AGR+16].

[8] We assume $n \geq 3$ since any 2-bit S-Box is linear/affine.

is to minimize the total number of non-linear operations and due to the same arguments given for MiMC in 10.3.2, it turns out that the best solution is given by the cubic S-Box. Roughly speaking, the main difference – when changing the S-Box – is about the security against algebraic attacks. An S-Box with a higher degree than the cubic one allows to reach the maximum degree much faster, hence a smaller number of rounds is potentially sufficient to guarantee security. At the same time, an S-Box with a higher degree requires more linear and non-linear operations in order to be computed. As a result, even if the number of rounds can be *potentially* decreased[9], the total number of (non-linear) operations basically does not change.

About the MixLayer. One possibility is to implement the MixLayer using a $t \times t$ MDS matrix. Such a matrix with elements in $GF(2^n)$ (or $GF(p)$ where $p \approx 2^n$) exists if the condition (see [MS78] for details)

$$\log_2(2t+1) \leq n$$

(or equivalently $t \cdot \log_2(2t+1) \leq N$) is satisfied. This implies a condition on the parameter t and n, that is[10]

$$n \geq \log_2 \left(\frac{2N}{\log_2(2N+1)} + 1 \right).$$

Given n and t, a possible way to construct an MDS matrix is using a *Cauchy matrix.* Let $x_i, y_i \in \mathbb{F}_{2^n}$ for $i = 1, ..., t$ s.t.

$$\forall i \neq j : \quad x_i \neq x_j \text{ and } y_i \neq y_j \text{ and } x_i \oplus y_j \neq 0$$

To fulfill these conditions, one can simply consider x_i s.t. the $t - \log_2(t)$ most significant bits are zero. Then, choosing $r \in \mathbb{F}_{2^n}$ s.t. the $t - \log_2(t)$ most significant bits are non zero, let $y_i = x_i \oplus r$. Let A be the Cauchy matrix defined by

$$A_{i,j} = \frac{1}{x_i \oplus y_j}.$$

It is possible to prove that A is MDS.

Other Possible Choices. For completeness, we mention that the MixLayer can be implemented by e.g. near-MDS, or more generally by any invertible matrix.

Definition 25. *A $t \times t$ matrix M is called a near-MDS matrix if*

$$\mathcal{B}_d(M) = \mathcal{B}_d(M^{-1}) = \mathcal{B}_l(M) = \mathcal{B}_l(M^{-1}) = t$$

where $\mathcal{B}_d(\cdot)$ and $\mathcal{B}_l(\cdot)$ denote respectively the differential and the linear branch number.

Lemma 10. *A $t \times t$ matrix M is near-MDS if and only if for any $1 \leq s \leq t-1$ each $s \times (s+1)$ and $(s+1) \times s$ submatrix of M has at least one $s \times s$ non-singular submatrix.*

By definition, the branch number of a near-MDS is smaller than the corresponding one of a MDS matrix. As a result, we expect than a bigger number of rounds are required to guarantee security against some attacks. On the other hands, a near-MDS can be much cheaper to implement than a MDS matrix, which could be crucial for some applications. Several lightweight linear diffusion layers from $t \times t$ near-MDS Matrices for $t \leq 9$ are proposed in [LW17].

[9] For completeness, we mention that this is not always the case. As concrete example, we refer to [GLR+19, App. E], where we study the security of HADESMIMC instantiated by the inverse S-Box.

[10] To get this result, work with sequential approximations. Let t be $t = t_0 + t_1 + t_2 + ...$, where $t_i \gg t_{i+1}$. Firstly, $t_0 = N$. Secondly, let $t = t_0 + t_1 = N \cdot (1 - \varepsilon)$, where $(1 - \varepsilon) \log_2(2N+1) \approx 1$. It follows that $t \simeq \frac{N}{\log_2(2N+1)}$, and so on.

Key Schedule. Let k be the secret key of size N, that is, $k = [k_0 \| k_1 \| ... \| k_{t-1}]$, where $\|$ denotes concatenation, and where the size of k_j is $|k_j| = n$ for each $0 \leq j < t$. We define the i-th round key $k^{(i)}$ for $0 \leq i \leq R$ (where R is the number of rounds) as follows. For the first round $i = 0$, the subkey is simply given by the whitening key, that is, $k^{(0)} = k$. For the next rounds, the subkeys are defined by a linear key schedule as

$$\forall i = 1, ..., R : \quad k^{(i)} = \hat{M} \cdot k^{(i-1)} + RC^{(i)},$$

where $RC^{(i)} \neq 0$ is a random constant and $\hat{M} \neq M$. W.r.t. M (the matrix used to define the MixLayer of each round), we require that

$$\hat{M}^i = \underbrace{\hat{M} \times \hat{M} \times \cdots \times \hat{M} \times \hat{M}}_{i \text{ times}}$$

has no zero coefficient[11] for $1 \leq i \leq R$, where R denotes the number of rounds.

This condition implies that *each word of each subkey $k^{(i)}$ (linearly) depends on all words of k.* As a result, even if the attacker guesses a certain number of words of a subkey $k^{(i)}$, they do not have information about other subkeys (more precisely, they cannot deduce any words of other subkeys).

Different Key Size. For completeness, we mention that it is also possible to consider keys of size different from N. E.g., for a key k' of size n, we define the subkeys as

$$\forall i = 0, ..., R : \quad k^{(i)} = [k' \| k' \| \cdots \| k'] \oplus RC^{(i)},$$

for a random constant $RC^{(i)}$. Even if a lower number of rounds for the case $|k'| = n$ is potentially possible, we impose it to be equal to the number of rounds for the case $|k| = N = n \cdot t$.

An Efficient Implementation of HadesMiMC

Like for LowMC, the fact that the non-linear layer is partial in R_P rounds can be exploited in order to reduce the amount of operations in each round R_P. Referring to [KPP+17, Sect. 3] and to [DKP+19], we recall here an equivalent representation of an SPN with partial non-linear layer for an efficient implementation.

Round Constants. In the description of an SPN, it is possible to swap the order of the linear layer and the round key addition as both operations are linear. The round key then needs to be exchanged with an equivalent one. For round key $k^{(i)}$, the equivalent one can be written as $\hat{k}^{(i)} = MC^{-1}(k^{(i)})$, where MC is the linear layer in the i-th round. If one works with partial non-linear layers, it is possible to use this property to move parts of the original round keys from the last round all the way through the cipher to the whitening key. To arrive at such a reduced variant, we work as following:

- First, we find an equivalent key that is applied before the affine layer by moving the round key through the affine layer.
- Then we split the round key in two parts, one that applies to the S-Box part of the non-linear layer and one that applies to the identity part of the non-linear layer. The key part that only applies to the non-linear layer part can now move further up where it is merged with the previous round key.
- Working in this way for all round keys, we finally end up with an equivalent representation in which round keys are only added to the output of the S-Boxes apart from one whitening key which is applied to the entire state after the first R_f rounds.

[11] If no matrix satisfies this condition, then the idea is to choose a matrix that minimizes the total number of zero coefficients.

Note that the round keys of this equivalent representation can still be calculated as linear functions of the master key.

This simplified representation can in certain cases also reduce the implementation cost of an SPN block cipher with a partial non-linear layer. For instance, the standard representation of HadesMiMC requires key matrices of total size $t \cdot n \cdot (R+1)$, where $R = R_P + R_F$ is the number of rounds. The optimized representation only requires $t \cdot n \cdot (R_F + 1) + n \cdot R_P$, thus potentially greatly reducing the amount of needed memory and calculation to produce the round keys.

Linear Layer. A similar trick can be used also for the matrix multiplication.

Focusing on the rounds with a single S-Box, let M be the $t \times t$ MDS matrix of the linear layer:

$$M = \left[\begin{array}{c|ccccc} M_{0,0} & M_{0,1} & M_{0,2} & \cdots & M_{0,t-1} & M_{0,t} \\ \hline M_{1,0} & & & & & \\ M_{2,0} & & & & & \\ \vdots & & & \hat{M} & & \\ M_{t-1,0} & & & & & \\ M_{t,0} & & & & & \end{array}\right] \equiv \left[\begin{array}{c|c} M_{0,0} & v \\ \hline w & \hat{M} \end{array}\right]$$

where $\hat{M}$ is a $(t-1) \times (t-1)$ MDS matrix (note that since M is MDS, every submatrix of M is also MDS), v is a $1 \times (t-1)$ matrix and w is a $(t-1) \times 1$ vector.

By simple computation, the following equivalence holds:

$$M = \underbrace{\left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \hat{M} \end{array}\right]}_{M'} \times \underbrace{\left[\begin{array}{c|c} M_{0,0} & v \\ \hline \hat{w} & I \end{array}\right]}_{M''}, \tag{12.1}$$

where

$$\hat{w} = \hat{M}^{-1} \times w$$

and I is the $(t-1) \times (t-1)$ identity matrix. Note that both M' and $M^{''}$ are two invertible matrices[12].

As for the round constants discussed previously, it is possible to use the property (12.1) in order *to swap the S-Box layer (formed by a single S-Box and $t-1$ identity functions) and the matrix multiplication with the matrix M'*. As a result, each linear part in the R_P rounds is defined only by a multiplication with a matrix of the form $M^{''}$, which is a *sparse matrix*, since $(t-1)^2 - (t-1) = t^2 - 3t + 2$ coefficients of $M^{''}$ are equal to zero (moreover, $t-1$ coefficients of $M^{''}$ are equal to one). It follows that this optimized representation – potentially – greatly reduces the amount of needed memory and calculation to compute the linear layer multiplication.

Remark. *Since we focus only on applications like PQ-Signature schemes and MPC, we do not define an hash function that exploits the design strategy proposed here. As future work, it could be interesting to study if a sponge construction instantiated with a permutation based on the* Hades *strategy (e.g.* HadesMiMC *with a fixed key) can be competitive for applications like SNARKs.*

12.3. Security Analysis

As for any new design, it is paramount to present a concrete security analysis. In the following, we provide an in-depth analysis of the security of the HadesMiMC family of block ciphers. Due to a lack of any method to ensure that an efficient cipher design is secure against all possible attacks, the

[12] First of all, $\det(M') = \det(\hat{M}) \neq 0$ since $\hat{M}$ is an MDS matrix, and so it is invertible. Secondly, $\det(M) = \det(M') \cdot \det(M^{''})$. Since $\det(M) \neq 0$ and $\det(M') \neq 0$, it follows that $\det(M^{''}) \neq 0$.

best option of determining a cipher's security is to ensure that the cipher is secure against all known attacks. Following this design strategy from the literature for new designs, we exploited this strategy also for our proposals. The number of rounds of HADESMIMC is then chosen in order to provide security against all known attack vectors.

We remark that security against statistical attacks is obtained exploiting the "Wide-Trail Strategy", that is, a certain number of rounds $R_F^{stat} = 2 \cdot R_f^{stat}$ with full S-Box layer is chosen in order to prevent statistical attacks. Then, more rounds both with full S-Box layer – that is, $R_F = R_F^{stat} + R' \geq R_F^{stat}$ (at least equal to the ones necessary for the "Wide-Trail Strategy" to work) – *and/or* with partial S-Box layer – that is, $R_P \geq 0$ – are added (if necessary) in order to guarantee security against all other possible attacks.

As already mentioned, rounds with partial S-Box layers are in general sufficient to increase the degree of the encryption/decryption function, which is the main factor that influences the cost of an algebraic attack. Thus, it would be natural to use rounds with partial S-Box layers in order to guarantee security against algebraic attacks. On the other hand, in general other factors play a crucial role in the cost of an algebraic attack. Thus, depending on the cost metric that we want to minimize, it makes sense to *choose the best ratio between rounds with full S-Box layer and rounds with partial S-Box layer in order to achieve both security and to minimize the cost metric.*

Important Remark. *Due to our target applications, we limit ourselves to provide the number of rounds necessary to guarantee security *only* in the following two scenarios:*

- HADESMIMC *instantiated over $\mathbb{F}_p$ (used for MPC application);*
- HADESMIMC *instantiated over $\mathbb{F}_{2^n}$ in the low-data scenario (used for applications like the PQ-signature scheme).*

We stress that this choice is motivated by the fact that we focus only on the scenarios that are useful for our applications. In Table 12.1, we present the minimum number of rounds with full S-Box layers R_F *or – if possible –* with partial S-Box layers R_P that are required to provide security of HADESMIMC instantiated with S-Box$(x) = x^3$ (and MDS matrix) in $\mathbb{F}_p$ against the corresponding attacks - no restriction on data complexity.

12.3.1. Main Points of Our Cryptanalysis Results

Here we would like to highlight the main points of our cryptanalysis results.

Number of Rounds. In the following, *given the number of rounds of a distinguisher which is independent of the secret key, we arbitrarily add (at least) 2 rounds to prevent key-guessing attacks.* This choice is motivated by the fact that – due to the key schedule and due to the MixLayer – it is not possible to skip more than a single round without guessing the entire key.

Statistical Attacks. As we show, 6 or 8 rounds with full S-Box layers are sufficient to protect HADESMIMC against all statistical attacks in the literature (that is, differential, linear, truncated/impossible differential, boomerang, ...).

Algebraic Attacks. Algebraic attacks exploit mainly the low degree of the encryption/decryption function in order to break the cipher. However, as already mentioned, other factors can influence the cost of such attacks. In particular:

Interpolation Attack. The goal of an interpolation attack is to construct the polynomial that describes the function: if the number of monomials is too big, then such a polynomial can not be

Table 12.1.: Minimum number of rounds with full S-Box layers $R_F = 2 \cdot R_f$ *and/or* with partial S-Box layers R_P necessary to provide security of HadesMiMC instantiated by respectively S-Box$(x) = x^3$ (and MDS matrix) in $\mathbb{F}_p$ against the corresponding attacks - no restriction on data complexity. We emphasize that the number of rounds necessary to prevent the interpolation attack are also sufficient to guarantee security against the higher-order differential attack (working in $\mathbb{F}_p$).

Attack	**Condition for Security**
Differential/Linear	$R_F \geq R_F^{stat} \equiv \begin{cases} 6 & \text{if } t+2 < 2 \cdot \lfloor \log_2(p) \rfloor \\ 8 & \text{if } t+2 \geq 2 \cdot \lfloor \log_2(p) \rfloor \end{cases}$
(MitM) Truncated Diff.	$R_F \geq 6$
Impossible Diff.	$R_F \geq 4$
Multiple-of-8/Mixture Diff.	$R_F \geq 4$
Boomerang Attack	$R_F \geq 6$
Integral	$R_F \geq 4$
Interpolation Attack	$R_F + R_P \geq R^{inter}(N,t) \equiv 5 + \lceil \log_3(p) \rceil + \lceil \log_3(t) \rceil$
GCD Attack	$R_P + R_F \geq 4 + \lceil \log_3(p) \rceil - \lfloor 2 \cdot \log_3(\log_2(p)) \rfloor$
Gröbner Basis	$\begin{cases} R_P + R_F \geq R^{1st-Grob}(N,t) \equiv 2 + \left\lceil \log_3(2) \cdot \left\lceil \frac{\log_2(p)}{2} + \log_2(t) \right\rceil \right\rceil \\ R_P + t \times R_F \geq R^{2nd-Grob}(N,t) \equiv \left\lceil \frac{N}{2 \cdot \log_2((2p-1)/3)} \right\rceil + \left\lceil \frac{N}{2 \cdot \log_2(27/4)} \right\rceil \\ R_F \geq R^{3rd-Grob}(N,t,R_P) \equiv 2 + \left\lceil \log_3(2) \cdot \left\lceil \frac{N}{2t+R_P} + 2 \cdot \log_2\left(1 + \frac{R_P}{2t}\right) \right\rceil \right\rceil \end{cases}$

constructed faster than via a brute force attack. A (lower/upper) bound of the number of different monomials can be estimated given the degree of the function. We show that – when the polynomial is dense – the attack complexity is approximately $\mathcal{O}(d^t)$, whereas d is the degree of the polynomial after r rounds. Since $d = 3^r$ for the cubic case, $\log_3(p) + \log_3(t)$ rounds with partial S-Box layers are necessary to guarantee security, where $\log_3(t)$ more rounds guarantee that the polynomial is dense. The cost of the attack does not change when working with rounds with full S-Box layers.

Finally, note that the degree of a function can also depend on its "representation". To give a concrete example, the function x^{-1} can be written as a function of degree $2^n - 2$ (namely, $x^{-1} \equiv x^{2^n-2}$) or using the "fraction representation" $1/x$ as introduced in [JK97], where both the numerator and the denominator are functions of degree at most equal to 1 – see also [GLR+19, App. E].

Gröbner Basis Attack. In a Gröbner basis attack, one tries to solve a system of non-linear equations that describe the cipher. The cost of such an attack depends obviously on the degree of the equations, but also on the number of equations and on the number of variables. We show that – when working with rounds with full S-Box layers – the attack complexity is approximately $\mathcal{O}((d/t)^t)$. If a partial S-Box layer is used in order to guarantee security against this attack, it could become more efficient to consider degree-3 equations for single S-Boxes. In this case, a higher number of rounds can be necessary in order to guarantee security against this attack.

To summarize, a round with a partial S-Box layer can be described by just 1 non-linear equation of degree d and $t-1$ linear equations, while a round with a full S-Box layer can be described by t non-linear equations of degree d. If the cost of the attack depends on other variables than just the degree (as in the case of a Gröbner basis attack), this fact can influence its final cost.

Higher-Order Diff. The higher-order differential attack exploits the property that given a function $f(\cdot)$ of algebraic degree δ, then $\bigoplus_{x \in V \oplus \phi} f(x) = 0$ if the dimension of the subspace V satisfies $dim(V) \geq \delta + 1$ (where the algebraic degree δ of a function $f(x) = x^d$ is given by $\delta = hw(d)$ where $hw(\cdot)$ is the hamming weight). If the algebraic degree is sufficiently high, then the attack does not work. As we are going to show, in the case in which HadesMiMC is instantiated over $\mathbb{F}_p$, security against interpolation attack implies security against this attack.

Other Attacks. *Related-Key Attacks.* The related-key attack model [Bih93] is a class of cryptanalytic attacks in which the attacker knows or chooses a relation between several keys and is given access to encryption/decryption functions with all these keys. We explicitly state that we do *not* make claims in the related-key model as we do not consider it to be relevant for the intended use case.

HADESMIMC-*Permutation: Security.* Since we do not require the indistinguishability of the permutation obtained by HADESMIMC with a fixed key from a "randomly drawn" permutation[13] in the practical applications considered in the following, we explicitly state that we do *not* make claims about the indistinguishability of the HADESMIMC-Permutation.

12.3.2. Security Analysis - Statistical Attacks

Here we consider security against statistical attacks. Since all statistical attacks that we are going to analyze work in the same way both for the case in which HADESMIMC is instantiated over $\mathbb{F}_p$ or over $\mathbb{F}_{2^n}$, we do not consider these two cases separately.

Differential Cryptanalysis

Differential cryptanalysis [BS90; BS93] and its variations are the most widely used techniques to analyze symmetric-key primitives. The differential probability of any function over the finite field $\mathbb{F}_{2^n}$ is defined as

$$Prob[\alpha \to \beta] := |\{x : f(x) \oplus f(x \oplus \alpha) = \beta\}|/(2^n).$$

Since the cubic function $f(x) = x^3$ is an almost perfect non-linear permutation (APN) [NK92], it has an optimal differential probability over a prime field or $\mathbb{F}_{2^n}$ (where n is odd). In other words, for this function the probability is bounded above by $2/2^n$ or $2/|\mathbb{F}_p|$.

As largely done in the literature, we claim that HADESMIMC is secure against differential cryptanalysis if each characteristic has probability at most 2^{-N}. In order to compute the minimum number of rounds to guarantee this, we work only with the rounds with full S-Box layers. In other words, we limit ourselves to work with a "weaker" version of the cipher defined as

$$R^{R_f} \circ L \circ R^{R_f}(\cdot), \tag{12.2}$$

where

- L is an *invertible linear layer* (which is the "*weakest*" possible assumption),
- $R(\cdot) = M \circ \text{S-Box} \circ ARK(\cdot)$ where S-Box$(\cdot)$ is a full S-Box layer (remember that M is an MDS matrix).

We are going to show that this "weaker" cipher is secure against differential cryptanalysis for

$$R_F^{stat} = \begin{cases} 6 & \text{if } t+2 < 2 \cdot \lceil \log_2(p) \rceil \\ 8 & \text{if } t+2 \geq 2 \cdot \lceil \log_2(p) \rceil \end{cases} \tag{12.3}$$

As a result, it follows that also HADESMIMC (instantiated with R_F rounds with full S-Box layers) is secure against such an attack. Indeed, if the linear layer L (which we only assume to be invertible) is replaced by R_P rounds of HADESMIMC, its security cannot decrease. *The same strategy is exploited in the following in order to prove security against all attacks in this subsection.*

[13] This basically corresponds to the known-key [KR07] or chosen-key models, where the attacker can have access or even choose the key(s) used, and where the goal is to find some (plaintext, ciphertext) pairs having a certain property with a complexity lower than what is expected for randomly chosen permutations.

In order to prove the result just given, we need a lower bound on the (minimum) number of active S-Boxes. Observe that the minimum number of "active" S-Boxes of a cipher of the form

$$R^s \circ L \circ R^r(\cdot) \equiv SB \circ \underbrace{M \circ SB}_{s-1 \text{ times}} \circ \underbrace{L'}_{\equiv L \circ M(\cdot)} \circ SB \circ \underbrace{M \circ SB}_{r-1 \text{ times}}(\cdot),$$

where $s, r \geq 1$, $R(\cdot)$ is a round with a full S-Box layer and where L' is an invertible linear layer, is at least[14]

$$\text{number } \textit{active} \text{ S-Boxes} \geq \underbrace{\big(\lfloor s/2 \rfloor + \lfloor r/2 \rfloor\big) \cdot (t+1)}_{\text{due to final/initial rounds}} + \big(s \text{ mod } 2\big) + \big(r \text{ mod } 2\big).$$

We emphasize that the (middle) linear $L'(\cdot) \equiv L \circ M(\cdot)$ plays *no* role in the computation of the previous number (remember that it has branch number equal to 2). Since at least $2 \cdot (t+1) + 1$ S-Boxes are active in the 5 middle rounds of $R^r \circ L \circ R^{5-r}(\cdot)$ for $1 \leq r \leq 4$, and since the maximum differential probability of the cubic S-Box is $DP_{max} = 2^{-n+1}$, each characteristic has probability at most

$$(2^{-n+1})^{2 \cdot (t+1)+1} = 2^{-N} \cdot 2^{-N-3n+2t+3} < 2^{-N},$$

since $[N + 3n = n \cdot (t+3)] > [2t + 3 = 2 \cdot (t + 3/2)]$, where $t + 3 > t + 3/2$ and $n \geq 3$. Finally, 2 more rounds guarantee that no differential attack can be set up by key guessing. Indeed, note that *(1st) given a partial round key, one has no information about the other round keys – due to the key schedule* – and *(2nd) 1 round is sufficient to provide full diffusion.*

Similarly, in the case in which $t + 2 < 2n$, it is sufficient to consider the 3 middle rounds to guarantee security against differential cryptanalysis. Indeed, each characteristic has probability $(2^{-n+1})^{t+2} = 2^{-N} \cdot 2^{-2n+t+2} < 2^{-N}$, since at least $t + 2$ S-Boxes are active. Again, 2 more rounds guarantee that no differential attack can be set up by key guessing.

Linear Cryptanalysis

Similar to differential attacks, linear attacks [Mat93] pose no threat to the HadesMiMC family of block ciphers instantiated with the same number of rounds previously defined for classical differential cryptanalysis. This follows from the fact that the cubic function is almost bent (AB), which means that its maximum square correlation is limited to 2^{-n+1} (see [Mat93] for details). As a result, it offers the best possible resistance against linear cryptanalysis much like an APN function provides optimal resistance against differential cryptanalysis.

For completeness, we remember a function $f(\cdot)$ is AB and/or APN if and only if its inverse $f^{-1}(\cdot)$ is AB and/or APN [CCZ98]. As a result, both the encryption and the decryption processes are secure against linear and differential cryptanalysis[15].

Truncated Differential

A variant of classical differential cryptanalysis is the truncated differential one [Knu94], in which the attacker can predict only part of the difference between pairs of texts.

We consider the "weaker" cipher described in (12.2) again. Focusing only on active/passive bytes (and not on the actual differences), there exist several differentials with probability 1 for a maximum of 1 round of HadesMiMC, e.g.

$$[\alpha, 0, ..., 0]^T \xrightarrow{R(\cdot)} M \times [\beta, 0, ..., 0]^T$$

[14] If $s = 2 \cdot s'$ is even, then the minimum number of active S-Boxes over $R^s(\cdot)$ rounds with full S-Box layer is $\lfloor s/2 \rfloor \cdot (t+1)$. Instead, if $s = 2 \cdot s' + 1$ is odd, then the minimum number of active S-Boxes over $R^s(\cdot)$ rounds with full S-Box layer is $\lfloor s/2 \rfloor \cdot (t+1) + 1$.

[15] Remember that if a matrix M is MDS, then also M^{-1} is MDS.

where α, β denote non-zero differences. Due to the next S-Box layer, the linear relations given by $M \times (\beta, 0, ..., 0)^T$ are destroyed in the next round. As a result, no probability-one truncated differential covers more than a single round.

For comparison, in the AES case it is possible to set up a 3-round truncated differential trails (which are independent of the S-Box) even if 2-round AES already provides full diffusion. The reason of this is that the AES mixing layer works only on part of the state, while in our case it works on the entire state (that is, 2 rounds of AES are necessary to provide full diffusion, while 1 round of HADESMIMC is sufficient)

To summarize, even we do not exclude the possibility to set up longer truncated differential trails (which depend or not on the details of S-Box), it seems hard to set up a truncated differential which is independent of the secret key for more than 2 rounds. As a result and due to the key schedule, we conjecture that 4 rounds with full S-Box layer makes HADESMIMC secure against this attack.

Differential Meet-in-the-Middle Attack

A possible way to extend (truncated) differential attacks over more rounds is using the Meet-in-the-Middle (MitM) technique[16]. The main idea is to split the cipher into two independent parts and use a time-memory trade-off for a more efficient attack. In more details, assume to split the cipher E into two parts $E(\cdot) = E_2 \circ E_1(\cdot)$. Roughly speaking, given a plaintext-ciphertext pair (p, c) obtained under the secret key K, the attacker partially guesses the secret key and computes

$$p \xrightarrow{E_1(\cdot)} \overrightarrow{v} \overset{?}{=} \overleftarrow{v} \xleftarrow{E_2(\cdot)} c.$$

If there is no match in the middle, it turns out that the guessed key is wrong.

Due to the truncated differential analysis just proposed and the fact that 1-round HADESMIMC provides full diffusion, we argue that 6 rounds of HADESMIMC with full S-Box layers are sufficient to guarantee security against this attack. For comparison, note that the best MitM attack on AES-128 covers 7 rounds [DF13], but 2 rounds of AES are necessary to guarantee full diffusion (instead of 1).

Impossible Differential

Impossible differential cryptanalysis was introduced by Biham *et al.* [BBS99] and Knudsen [Knu98]. This cryptanalytic technique exploits differentials occurring with probability 0.

In the following, we focus only on impossible differentials which are independent of the S-Box details, i.e., we do not consider the actual differences but only active/passive words. To find them, we use the possible transitions of the linear layer combined with the fact that the S-Box is a bijection. We found that the longest impossible differential (in this class) only spans 2 rounds, e.g.,

$$\begin{bmatrix} \alpha \\ 0 \\ ... \\ 0 \end{bmatrix} \xrightarrow[\text{prob. 1}]{R(\cdot)} \begin{bmatrix} M_{0,0} \cdot \beta \\ M_{1,0} \cdot \beta \\ ... \\ M_{t-1,0} \cdot \beta \end{bmatrix} \neq \begin{bmatrix} \gamma \\ 0 \\ ... \\ 0 \end{bmatrix} \xleftarrow[\text{prob. 1}]{R^{-1}(\cdot)} \begin{bmatrix} M_{0,0} \cdot \delta \\ M_{1,0} \cdot \delta \\ ... \\ M_{t-1,0} \cdot \delta \end{bmatrix}$$

for $\alpha, \beta, \gamma, \delta \neq 0$ (note that no coefficient of M is equal to zero since M is an MDS matrix). As a result and due to the key schedule, it turns out that 6 rounds with full S-Box layers makes HADESMIMC secure against this attack.

Note that it is possible to compare this result with a similar one on AES, where the best known impossible differential is also in this class (of impossible differentials) and spans four rounds [BK01].

[16] We refer to Sect. 12.3.3 for a discussion about the security against "algebraic" Meet-in-the-Middle Attacks - here we focus on differential MitM attacks.

Boomerang Attack

In boomerang attacks [Wag99], good partial differential characteristics that cover only part of the cipher can be combined to attack ciphers that might be immune to standard differential cryptanalysis. In these attacks, two differential characteristics are combined, one that covers the first part of the cipher and another that covers the second part. If both have about the same probability, the complexity corresponds roughly to the inverse of the product of the square of each of their probabilities [Wag99].

To calculate the number of rounds sufficient to ensure that no good boomerang exists, we determine the number of rounds after which we cannot separate the cipher into two parts and find a differential for each such that the product of their probabilities is less than $2^{-N/2}$. Exploiting the analysis proposed before, it turns out that 6 rounds with full S-Boxes are sufficient for this goal.

Multiple-of-n and Mixed Differential Cryptanalysis

The "Multiple-of-8" distinguisher [GRR17] was proposed at Eurocrypt 2017 by Grassi *et al.* as the first 5-round secret-key distinguisher for AES that exploits a property which is independent of the secret key and of the details of the S-Box. It is based on a new structural property for up to 5 rounds of AES: by appropriate choices of a number of input pairs it is possible to make sure that the number of times that the difference of the resulting output pairs lie in a particular subspace is always a multiple of 8. The input pairs of texts that satisfy a certain output difference are related by linear/differential relations. Such relations are exploited by a variant of such a distinguisher, called the "mixture differential" distinguisher [Gra18b] proposed at FSE/ToSC 2019.

Regarding HadesMiMC, it is possible to set up such distinguishers on 2 rounds only. In particular, consider a set of texts with $2 \leq s \leq t$ active words (and $t - s$ constants words). The number of pairs of texts that satisfy an (arbitrary) output truncated differential is always a multiple of 2^{s-1}. Moreover, the relations of the input pairs of texts exploited by mixture differential cryptanalysis are known.

The proofs of these two properties are analogous to the ones proposed in [GRR17; Gra18b] and recently in [BCC19]. E.g., consider two texts T^1 and T^2 of the form

$$T^1 = c' \oplus \begin{bmatrix} x_0 & x_1 & 0 & ... & 0 \end{bmatrix}^T, \qquad T^2 = c' \oplus \begin{bmatrix} y_0 & y_1 & 0 & ... & 0 \end{bmatrix}^T$$

for some constant c' and where $x_i \neq y_i$ for $i = 0, 1$. After one round, the difference in each word is of the form

$$M_0 \cdot [\text{S-Box}(x_0 \oplus c_0) \oplus \text{S-Box}(x_1 \oplus c_1)] \oplus M_1 \cdot [\text{S-Box}(y_0 \oplus c_0) \oplus \text{S-Box}(y_1 \oplus c_1)],$$

where M_0, M_1 depend on the MixLayer and c_0, c_1 depend on the secret key. By simple observation, the same output difference is given by the pair of texts

$$\hat{T}^1 = c' \oplus \begin{bmatrix} y_0 & x_1 & 0 & ... & 0 \end{bmatrix}^T, \qquad \hat{T}^2 = c' \oplus \begin{bmatrix} x_0 & y_1 & 0 & ... & 0 \end{bmatrix}^T.$$

Combining this result with a 1-round truncated differential with prob. 1, it is possible to set up a multiple-of-n distinguisher (where $n = 2^{s-1}$) and a mixture differential one on 2 rounds of HadesMiMC. As a result and due to the key schedule, it turns out that 6 rounds with full S-Box layers make HadesMiMC secure against these attacks.

Biclique Cryptanalysis

Biclique cryptanalysis [BKR11] can be viewed as an improvement of classical MitM attacks. It improves the complexity of exhaustive search by computing only a part of the encryption algorithm. The improved factor – often evaluated by the ratio of the number of S-Boxes involved in the partial computation to all S-Boxes in the cipher – *can* be relatively big when the number of rounds in the cipher is (very) small. Since we do not think that improving the exhaustive search by a small factor will turn into a serious vulnerability in future, HADESMIMC is not designed to resist biclique cryptanalysis with small improvement.

Integral/Square Attack

Integral cryptanalysis is a technique first applied on SQUARE [DKR97] and is particularly efficient against block ciphers based on substitution-permutation networks, like AES or HADESMIMC.

The idea is to study the propagation of sums of values. For the case of HADESMIMC, it is possible to set up an integral distinguisher over two rounds, e.g.

$$\begin{bmatrix} A \\ C \\ \dots \\ C \end{bmatrix} \xrightarrow{\text{S-Box}(\cdot)} \begin{bmatrix} A \\ C \\ \dots \\ C \end{bmatrix} \xrightarrow{M(\cdot)} \begin{bmatrix} A \\ A \\ \dots \\ A \end{bmatrix} \xrightarrow{\text{S-Box}(\cdot)} \begin{bmatrix} A \\ A \\ \dots \\ A \end{bmatrix} \xrightarrow{M(\cdot)} \begin{bmatrix} B \\ B \\ \dots \\ B \end{bmatrix}$$

where A denotes an active word, C a constant one and B a balanced one[17]. As a result and due to the key schedule, it turns out that 6 rounds with full S-Box layers make HADESMIMC secure against this attack.

Invariant Subspace Attack

The invariant subspace attack [LAAZ11] makes use of affine subspaces that are invariant under the round function. As the round key addition translates this invariant subspace [BCLR17], ciphers exhibit weak keys when all round keys are such that the affine subspace stays invariant including the key addition. Therefore, those attacks are mainly an issue for block ciphers that use identical round keys. In our case, the non-trivial key schedule already provides a good protection against such attacks for a larger number of rounds.

12.3.3. Security Analysis - Algebraic Attacks

Interpolation Attack

One of the most powerful attacks against HADESMIMC is the interpolation attack, introduced by Jakobsen and Knudsen [JK97] in 1997.

The strategy of the attack is to construct a polynomial corresponding to the encryption function without knowledge of the secret key. Let $E_k : \mathbb{F}_{2^N} \to \mathbb{F}_{2^N}$ be an encryption function. For a randomly fixed key k, the interpolation polynomial $P(x)$ representing $E_k(x)$ can be constructed[18] using e.g. the Vandermonde matrix - *cost* approximately of $\mathcal{O}(t^2)$ - or the Lagrange's theorem - *cost* approximately of $\mathcal{O}(t \cdot \log t)$, where x is the indeterminate corresponding to the plaintext. If an adversary can construct such an interpolation polynomial without using the full code-book, then she can potentially used it to set up a *forgery attack*. This can be exploited both as a secret-key distinguisher and in a scenario in which there is no key and/or secret material (e.g., in the hash scenario).

This method can also be extended to a key-recovery attack. The attack proceeds by simply guessing the key of the final round, decrypting the ciphertexts and constructing the polynomial for $r-1$ rounds[19]. With one extra (plaintext, ciphertext) pair, the attacker checks whether the polynomial is correct. The data cost of the attack is well approximated by the number of texts necessary to construct the interpolation polynomial.

[17] For completeness, we recall that given a set of texts $\{x_i\}_{i\in I}$, the word x^j is *active* if $x_i^j \neq x_l^j$ for each $i \neq l$, constant if $x_i^j = x_l^j$ for each i, l, and balanced if $\bigoplus_i x_i^j = 0$.

[18] We refer to Sect. 11.3.1 for more details about these two strategies.

[19] The "hidden" assumption is that *the cost to construct such a polynomial is smaller than the cost of an encryption.* If this assumption does not hold, then the cost of the attack is bigger than the cost of a brute-force attack.

Case: S-Box$(x) = x^3$**.** Considering HADESMIMC, by simple computation and since the S-Box is the cubic function, the degree of each word after r rounds is 3^r. However, since[20] all mixing terms of (total) degree 3^d appear at round 3^{d+1}, we assume in the following that the degree of each word after r rounds is 3^{r-1}. In particular, *since in each round at least one S-Box is applied and since the affine layer does not change the algebraic degree, the algebraic degree of one round is three as well.* In other words, one S-Box per round (together with a "good" affine layer) is sufficient to increase the degree of each word. For this reason, in the following we consider a *weaker cipher in which each round contains only a single S-Box. If such a cipher is secure against the interpolation attack, then our design is also secure*[21] (more S-Boxes per round do not decrease the security).

Assuming the interpolation polynomial is not sparse, a (rough) estimation of the number of monomials of the interpolation polynomial (and so of the complexity of the attack) is given by

$$(3^{r-1}+1)^t \geq 3^{(r-1)\cdot t},$$

since after r rounds there are t words each of degree *at least* 3^{r-1}. As a result, by requiring that the number of monomials is equal to the full codebook $3^{(r-1)\cdot t} \simeq p^t$ (that is $3^{r-1} \simeq p$), the number of rounds must be at least $r \simeq 1 + \log_3(p)$.

Actually, the previous rough estimation of the number of rounds does not guarantee that the interpolation polynomial is not sparse. As showed in details[22] in [GLR+19, App. D], since the cipher works over a finite field with characteristic p and due to the specific algebraic structure of the cubic function, this problem can be avoid by adding $\lceil \log_3(t) \rceil$ more rounds.

Finally, a MitM variant of the interpolation attack can be performed. To thwart this variant and due to the high degree of S-Box$^{-1}(x) = x^{1/3} = x^{(2p-1)/3}$ (remember that $p = 2 \mod 3$ in order to guarantee that x^3 is invertible), it is sufficient to add 2 rounds[23]. Finally, we add 2 more rounds to prevent key-guessing attacks. As a result, the total number of rounds R must satisfy [24]

$$R = R_P + R_F \geq R^{inter}(N,t) \equiv 5 + \lceil \log_3(p) \rceil + \lceil \log_3(t) \rceil \tag{12.4}$$

to thwart the interpolation attack.

GCD Attack

As for MiMC [AGR+16], the Greatest Common Divisors (GCD) attack strategy also applies to HADESMIMC. In particular, given more than one known (plaintext, ciphertext) pair or working on the output of each S-Box of a single (known) pair, it is possible to construct their polynomial representations and compute their polynomial GCD to recover a multiple of the key. Note that this is a known-plaintext attack, and not a chosen-plaintext one.

Denote by $E(k,x)$ the encryption of x under key k. For a pair $(x,y) \in \mathbb{F}_{2^N}$, $E(K,x) - y$ denotes a univariate polynomial in $\mathbb{F}_q[K]$ corresponding to (x,y). Note that in our case the polynomial $E(K,x) - y$ can be constructed conceptually easily from the encryption process, but writing down $E(K,x) - y$ becomes computationally expensive as the number of rounds increases. Indeed, writing

[20] Note that after the first round not all words of degree 3 appear. Indeed, the input of each S-Box in the first round is composed of a single word, which means that after the first round there is no *non-linear* mixing of different words. Similarly, all mixing terms of (total) degree 3^d appears at round 3^{d+1}.

[21] In the case of partial S-Box layers, it could *potentially* be possible to skip a certain number of rounds by a proper choice of the input texts (e.g. by having no active S-Box). However, we do not care about this property here, since – due to the HADES strategy – at least the first and the last 3 rounds are going to have a full S-Box layer, which guarantees full diffusion.

[22] ***Remark.*** *Since I did not work on such result – it was done by Reinhard Lüftenegger, I limit myself to refer to [GLR+19, App. D] for all details.*

[23] Note that $\log_{(2p-1)/3}(p) \leq 2$ for each $p \geq 5$.

[24] We emphasize that *in this analysis we do not take into account the cost of constructing the interpolation polynomial, which is (in general) non-negligible.*

down $E(K, x) - y$ requires not only large computational resources but also an exponential (in r) amount of memory. Consider now two such polynomials $E(K, x_1) - y_1$ and $E(K, x_2) - y_2$, with $y_i = E(k, x_i)$ for $i = 1, 2$ and for a fixed but unknown key k. It is clear that these polynomials share $(K - k)$ as a factor. Indeed, with high probability the greatest common divisor will be $(K - k)$. Thus, by computing the GCD of the two polynomials, we can find the value of k. As we are going to show, this attack is less efficient than e.g. the interpolation attack. However, we remark that *this is one of the few attacks that applies in the low-data scenario*, considered in one of the following applications (i.e., post-quantum signatures). In particular, a single input-output pair is sufficient to compute the required polynomial $E(K, x) - y$, and the GCD can be computed among the output of two different S-Boxes of the final round. What about the complexity? It is well-known that the complexity for finding the GCD of two polynomials of degree d is $\mathcal{O}\left(M(d) \log_2 d\right)$, where $M(d)$ is the cost of multiplying two degree-d polynomials. The best (known) complexity for $M(d)$ is $\mathcal{O}(d \log_2 d)$ using an FFT. Thus, we expect a GCD computation to cost $\mathcal{O}\left(d \log_2^2 d\right)$, where the hidden constant is greater than 1. In order to estimate the computational cost of such an attack, we have to estimate the degree of K in $E(K, x) - y$, which depends on the number of rounds r.

Case: S-Box$(x) = x^3$. To set up the attack, the attacker first guesses $t - 1$ words of the key in order to construct an univariate polynomial. Since the complexity of the attack depends on the degree and since one S-Box per round (together with an affine layer) is sufficient to increase the degree of each word, we can focus only on the rounds R_P with a single S-Box. If such a cipher is secure against interpolation attack, also our design is secure (more S-Boxes per round do not decrease the security). As a result, after $r > 1$ rounds, the degree d is well estimated by 3^{r-1}. Thus, to derive an estimate for the required number of rounds, we target $d \log_2^2 d \approx p$, which implies[25]

$$r \geq \log_3 p + 1 - 2 \log_3(\log_2 p).$$

To thwart a MitM variant of this attack, it is sufficient that add 1 round. Finally, we add two rounds to prevent key-guessing attack. As a result, the total number of rounds must be

$$R_P + R_F \geq 4 + \lceil \log_3 p \rceil - 2 \lfloor \log_3(\log_2 p) \rfloor .$$

Gröbner Basis Attack

The natural generalization of GCDs is the notion of Gröbner basis [BKW93]. The attack proceeds like the GCD attack with the final GCD computation replaced by a Gröbner basis computation. Analogous to the GCD above and the interpolation analysis in the following, 1 S-Box per round is sufficient to prevent this attack (since it basically depends on the degree of the encryption function, which is independent of the number of S-Boxes per round).

For generic systems, the complexity of computing a Gröbner basis for a system of $\mathfrak{N}$ polynomials f_i in $\mathfrak{V}$ variables is

$$\mathcal{O}\left(\binom{\mathfrak{V} + D_{reg}}{D_{reg}}^{\omega}\right)$$

operations over the base field $\mathbb{F}$[BFP12], where D_{reg} is the *degree of regularity* and $2 \leq \omega < 3$ is the linear algebra constant. We note that the memory requirement of these algorithms is of the same order as the running time. The degree of regularity depends on the degrees of the polynomials d and the number of polynomials $\mathfrak{N}$. When $\mathfrak{V} = \mathfrak{N}$, we have the simple closed form

$$D_{reg} := 1 + \sum_{i=0}^{\mathfrak{V}-1} (d_i - 1), \qquad (12.5)$$

[25] Note that the solution of $y = x \cdot \log^2(x)$ is well approximated by $x = y / \log^2(y)$.

where d_i is the degree of the i-th polynomial f_i in the polynomial system we are trying to solve. In the over-determined case, i.e., $\mathfrak{V} < \mathfrak{N}$, the degree of regularity can be estimated by developing the Hilbert series of an ideal generated by generic polynomials $\langle f_0, \ldots, f_{\mathfrak{N}-1}\rangle$ of degrees d_i (under the assumption that the polynomials behave like generic systems). Closed form formulas for D_{reg} are known for some special cases, but not in general.

Low-Data Case – 1 (plaintext, ciphertext) pair. Let's start by analyzing the security of HadesMiMC against a Gröbner basis attack in the case in which the attacker has access to a single input/ouput pair. This attack in this scenario is very similar to the GCD attack just described, but it is less competitive.

In the case in which the attacker has access to a single known (plaintext, ciphertext) pair – denoted by $p, c \in (\mathbb{F}_{2^n})^t$ where $p \equiv (p_0, ..., p_{t-1})$ and $c \equiv (c_0, ..., c_{t-1})$ – the system is described by t *equations* of the form $c_i = f_i(p_0, ..., p_{t-1}, k_0, ..., k_{t-1})$ for $i = 0, ..., t-1$ and in t *variables* $k_0, ..., k_{t-1}$ (remember that the key schedule is linear). Using formula (12.5) in order to compute the degree of regularity for this case, it follows that $D_{reg} = 1 + t \cdot (d-1) \approx t \cdot 3^r$ where $d \simeq 3^r$ is the (approximately) degree after of each function f_i after r rounds. Thus, setting $\omega = 2$ with the hidden constant ≥ 1, the overall complexity becomes

$$\left[\binom{t + t\cdot 3^r}{t \cdot 3^r}\right]^2 \geq \left(\frac{(t\cdot 3^r)^t}{t!}\right)^2 \geq \left(\frac{t \cdot 3^r}{t}\right)^{2t} = (3^r)^{2t}$$

where $x! \leq x^x$ for all $x \geq 1$, and where $\prod_{i=1}^{s}(x+i) \geq x^s$. As a result, $r \geq 2 + \frac{\log_3(2)}{2} \cdot \log_2(p)$ are sufficient to prevent the attack.

Generic Case. Let's now consider the case in which the attacker has access to many (plaintext, ciphertext) pairs – the cost of the attack (potentially) decreases if the attacker has access to more than a single (plaintext, ciphertext) pair of texts. In this case, given at most $2^N - 1$ (plaintext, ciphertext) pairs – each one denoted by $p, c \in (\mathbb{F}_{2^n})^t$, where $p \equiv (p_0, ..., p_{t-1})$ and $c \equiv (c_0, ..., c_{t-1})$, the system is described by at most $\mathfrak{N} = t \cdot (2^N - 1)$ equations of the form $c_i = f_i(p_0, ..., p_{t-1}, k_0, ..., k_{t-1})$ in $\mathfrak{V} = t$ *variables* $k_0, ..., k_{t-1}$ (remember that the key schedule is linear). In this over-determined case $\mathfrak{N} > \mathfrak{V}$, there is no closed formula to compute D_{reg}. By definition, the degree of regularity is defined as the index of the first non-positive coefficient in

$$H(z) = \frac{\prod_{i=1}^{n_e}(1 - z^{d_i})}{(1-z)^{n_v}} = \frac{(1 - z^{3^r})^{n_e}}{(1-z)^{n_v}} = (1 - z^{3^r})^{n_e - n_v} \cdot (1 + z + z^2)^{n_v},$$

where n_e is the number of equations, n_v is the number of variables, and $d_i = 3^r$ is the degree of the i-th equation. By simple observation, the index of the first non-positive coefficient can not be smaller than $d = 3^r$, since $(1 + z + z^2)^{n_v}$ contains only positive terms (note that $n_e > n_v$).

Depending on parameter choices, the hybrid approach [BF09; BFP12] – which combines exhaustive search with Gröbner basis computations – may lead to a somewhat reduced cost. Following [BF09; BFP12], guessing $\kappa \leq t$ components of the key leads to a complexity of

$$\mathcal{O}\left(p^\kappa \cdot \binom{t - \kappa + D'_{reg}}{D'_{reg}}^{\omega}\right) \tag{12.6}$$

where $D'_{reg} \leq D_{reg}$ is the degree of regularity for the system of equations after substituting κ variables with their guesses.

It follows that to prevent Gröbner basis attacks, the minimum number of rounds r must satisfy $p^\kappa \cdot \binom{t-\kappa+D'_{reg}}{D'_{reg}}^{\omega} \geq p^t$, for all $0 \leq \kappa \leq t-2$ and where the degree of regularity $D'_{reg} = \mathcal{O}(d) \approx 3^r$. For our parameter choices, the expression (12.6) is minimized for $\kappa = 0$, which implies that

$$\binom{t+d}{d} = \frac{1}{t!} \cdot \prod_{i=1}^{t} (d+i) \geq \frac{d^t}{t!} \geq \left(\frac{d}{t}\right)^t = 2^{t \log_2(d/t)}$$

where $x! \leq x^x$ for each $x \geq 1$. Setting $\omega = 2$, we obtain $2t \log_2(d/t) \approx n \cdot t$ and

$$r \geq 2 + \log_3(2) \cdot \big(\log_2(p)/2 + \log_2(t)\big), \tag{12.7}$$

where 2 rounds are added to thwart the Meet-in-the-Middle version of the attack (note that the degree of the S-Box in the decryption direction is $(2p-1)/3$). As a result, $R \geq \big\lceil \log_3(2) \cdot \big(\log_2(p)/2 + \log_2(t)\big)\big\rceil + 2$ rounds are sufficient to protect the cipher from this attack. Note that the analysis just proposed is independent of the fact whether the rounds contain a full or a partial S-Box layer.

Round with Partial S-Box layer – First Alternative Strategy. The strategy just described is not the only possible one in order to set up a Gröbner Basis Attack. In particular, each equation of degree 3^r can be re-written in a different way, e.g. as r equations each one of degree 3. *Even if this allows to reduce D_{reg}, the introduction of new intermediate variables does not lead – in general – to a reduced solving time* (remember that the cost of a Gröbner basis depends both on the number of variables and of the degree of the system of equations that we are trying to solve).

As showed in the following, this second strategy does not outperform the one given before if rounds with full S-Box layer are used to guarantee security against Gröbner basis attack. Thus, $R_F^{Grobner} \geq \big\lceil \log_3(2) \cdot \big(\log_2(p)/2 + \log_2(t)\big)\big\rceil + 2$ rounds with full S-Box layer are sufficient to provide security against this attack.

The situation is different when one exploits rounds with partial S-Box layer in order to guarantee security. Using the strategy just proposed, the middle rounds with partial S-Box layer can be described by R_P variables and R_P equations of degree 3. In total, the number of variables and equations in the low-data case – 1 (plaintext, ciphertext) pair – is $R_F \cdot t + R_P - \kappa$ (where – as before – the best attack can be achieved without guessing any key word – that is, $\kappa = 0$), which means a cost of

$$\left[\binom{(R_F \cdot t + R_P) + (1 + 2 \cdot (R_F \cdot t + R_P))}{1 + 2 \cdot (R_F \cdot t + R_P)}\right]^2 \approx \left[\binom{3 \cdot (R_F \cdot t + R_P)}{2 \cdot (R_F \cdot t + R_P)}\right]^2 \approx \left(\frac{27}{4}\right)^{2(R_F \cdot t + R_P)},$$

and $R_F \cdot t + R_P \approx \frac{N}{2 \cdot (\log_2(27) - 2)}$. Working in the same way, the cost of the attack in the decryption direction (remember that S-Box$(x) = x^{(2p-1)/3}$) turns out to be

$$\left[\binom{(R_F \cdot t + R_P) + (1 + [(2p-4)/3] \cdot (R_F \cdot t + R_P))}{R_F \cdot t + R_P}\right]^2 \approx \left(\frac{2p-1}{3}\right)^{2(R_F \cdot t + R_P)},$$

It follows that

$$R_F \cdot t + R_P \approx \left\lceil \frac{N}{2 \cdot (\log_2(27) - 2)} \right\rceil + \left\lceil \frac{N}{2 \cdot (\log_2(2p-1) - \log_2(3))} \right\rceil,$$

in order to thwart the MitM version of the attack.

When working in the full-data case, the number of equations and variables can be increased by using multiple (plaintext, ciphertext) pairs. In particular, the key variables stay the same, while additional intermediate variables have to be introduced for each pair. Let $\mathfrak{D}$ denote the amount of data used by the attacker (obviously, $1 \leq \mathfrak{D} \leq 2^N - 1$). This means that the number of equations differs from the number of variables when using the same attack strategy together with multiple pairs. Therefore, we have to use the Hilbert series in order to determine the degree of regularity, which is the index of the first non-positive coefficient in

$$H(z) = \frac{\prod_{i=1}^{n_e}(1-z^{d_i})}{(1-z)^{n_v}} = \frac{(1-z^3)^{n_e}}{(1-z)^{n_v}} = (1-z^3)^{n_e-n_v} \cdot (1+z+z^2)^{n_v},$$

where n_e is the number of equations, n_v is the number of variables, and $d_i = 3$ is the degree of the i-th equation. When increasing the number of (plaintext, ciphertext) pairs $\mathfrak{D}$, both n_e and n_v increase. In our case:

$$n_e = \mathfrak{D} \cdot (t \cdot R_F + R_P) \qquad \text{and} \qquad n_v = t + \mathfrak{D} \cdot ((R_F - 1) \cdot t + R_P).$$

Note that $n_e - n_v = t \cdot (\mathfrak{D} - 1)$ and that $(1+z+z^2)^{n_v}$ contains only positive terms. Since

$$(1-z^3)^{n_e-n_v} = 1 - t \cdot (\mathfrak{D}-1) \cdot z^3 + ...$$

it follows that the index of the first non-positive coefficient must be at least 3, which means $D_{reg} \geq 3$. Unfortunately, this estimation is too pessimistic in order to derive a useful approximation of the number of rounds necessary to guarantee security. E.g., using $D_{reg} \approx 3$, the total cost of the attack is given by

$$\left[\binom{3+t+\mathfrak{D}\cdot((R_F-1)\cdot t+R_P)}{t+\mathfrak{D}\cdot((R_F-1)\cdot t+R_P)}\right]^2 \geq \left[\frac{(1+t+\mathfrak{D}\cdot((R_F-1)\cdot t+R_P))^3}{3!}\right]^2,$$

which means we need $R_P \approx \mathcal{O}(2^{N/6})$ in order to guarantee security.

However, by practical experiments, it turns out that *a much smaller number of rounds is sufficient for this scope*. Due to such practical results[26], we *conjecture* that

$$R_F \cdot t + R_P \approx \left\lceil \frac{N}{2 \cdot (\log_2(27) - 2)} \right\rceil + \left\lceil \frac{N}{2 \cdot (\log_2(2p-1) - \log_2(3))} \right\rceil$$

rounds are sufficient in order to protect HadesMiMC from this Gröbner basis attack strategy proposed here.

Finally, let us briefly analyze the case $R_P = 0$, which corresponds to the case in which the security against Gröbner basis attack is guaranteed by rounds with full S-Box layer. In this case, the previous inequality reduces to

$$R_F \geq \left\lceil \frac{\log_2(p)}{2 \cdot (\log_2(27) - 2)} \right\rceil + \left\lceil \frac{\log_2(p)}{2 \cdot (\log_2(2p-1) - \log_2(3))} \right\rceil \approx 1 + \left\lceil \frac{\log_2(p)}{2 \cdot (\log_2(27) - 2)} \right\rceil.$$

Since the generic strategy described before requires $R_F \geq 2 + \log_3(2) \cdot \big(\log_2(p)/2 + \log_2(t)\big)$ rounds in order to guarantee security, the attack strategy proposed here cannot outperform it (note that $2 \cdot \log_2(27/4) \geq 2 \cdot \log_2(3)$).

Round with Partial S-Box layer – Second Alternative Strategy. Another strategy can be applied as well. Let us work for simplicity in the encryption direction (similar results work in the decryption one). Given an input $x = (x_0, ..., x_{t-1})$, the output of the first R_f rounds with full S-Box layer can be described by t equations of degree 3^{R_f}, where $R_f = R_F/2$. Then, working round per round, the output of each round with partial S-Box layer is described by 1 non-linear equation of degree 3 and $t-1$ linear equations. Finally, t more equations of degree 3^{R_f} describes the relations between the output of the rounds with partial S-Box layer and the input of the final rounds with full S-Box layer. As a result, in the case of a single input/ouput pair, one works with

[26] ***Remark.*** *Since I did not work on such result – it was done by Markus Schofnegge, I limit myself to refer to [GLR+19, App. D] for all details.*

- $2t$ equations of degree 3^{R_f} and R_P equations of degree 3;
- t variables that describe the key and $t + R_P$ variables that describe the internal state of the texts.

Since the number of variables is equal to the number of equations, it follows that

$$D_{reg} = 1 + 2t \cdot (3^{R_F/2} - 1) + 2 \cdot R_P.$$

Setting $\omega = 2$, the cost of this attack is well described by

$$\left[\binom{(2t + R_P) + [1 + 2t \cdot (3^{R_F/2} - 1) + 2 \cdot R_P]}{2t + R_P}\right]^2 \approx \left[\binom{2t + 3 \cdot R_P + 2t \cdot 3^{R_F/2}}{2t + R_P}\right]^2 \geq$$

$$\geq \left(\frac{1 + 2 \cdot R_P + 2t \cdot 3^{R_F/2}}{2t + R_P}\right)^{2R_P+4t} \geq \left(\frac{2t \cdot 3^{R_F/2}}{2t + R_P}\right)^{2R_P+4t}.$$

It follows that HADESMIMC is secure if

$$R_F \geq 2 + \log_3(2) \cdot \left(\frac{N}{2t + R_P} + 2 \cdot \log_2(2t + R_P) - 2 \cdot \log_2(2t)\right),$$

where 2 more rounds have been added in order to thwart Meet-in-the-Middle versions of this attack.

The strategy is similar when one works with more (plaintext, ciphertext) pairs. As for the previous strategy, we found that *in all cases that we practically tested, the computational complexity is minimized when the attacker uses a single (plaintext, ciphertext) pair.* For this reason, we *conjecture* that HADESMIMC is secure against this version of the Gröbner basis attack if the following inequality

$$R_F \geq 2 + \log_3(2) \cdot \left(\frac{N}{2t + R_P} + 2 \cdot \log_2(2t + R_P) - 2 \cdot \log_2(2t)\right)$$

is satisfied.

Finally, let us briefly analyze the case $R_P = 0$, which corresponds to the case in which the security against Gröbner basis attack is guaranteed by rounds with full S-Box layer. In this case, the previous inequality reduces to $R_F \geq 2 + \log_3(2) \cdot \log_2(p)/2$. Since the strategy described in Sect. 12.3.3 requires $R_F \geq 2 + \log_3(2) \cdot \big(\log_2(p)/2 + \log_2(t)\big)$ rounds in order to guarantee security, the attack strategy proposed here cannot outperform it.

Gröbner Basis Attack – Recap. In conclusion, we claim HADESMIMC is secure against Gröbner basis attack if the following inequality is satisfied

$$R_F \geq R_F^{Grober}(N, t) \equiv 2 + \lceil \log_3(2) \cdot (\log_2(p)/2 + \log_2(t)) \rceil$$

or if the number of rounds $R_P \geq 1$ and R_F satisfy the following inequalities

$$\begin{cases} R_P + R_F \geq R^{1st-Grob}(N, t) \equiv 2 + \lceil \log_3(2) \cdot (\log_2(p)/2 + \log_2(t)) \rceil \\ R_P + t \times R_F \geq R^{2nd-Grob}(N, t) \equiv \lceil N/[2 \cdot \log_2((2p-1)/3)] \rceil + \lceil N/[2 \cdot \log_2(27/4)] \rceil \\ R_F \geq R^{3rd-Grob}(N, t, R_P) \equiv 2 + \log_3(2) \cdot \left(\frac{N}{2t+R_P} + 2 \cdot \log_2(2t + R_P) - 2 \cdot \log_2(2t)\right) \end{cases} \tag{12.8}$$

Higher-Order Differential Attack

A well-known result from the theory of Boolean functions is that if the algebraic degree of a vectorial Boolean function $f(\cdot)$ (like a permutation) is d, then the sum over the outputs of the function applied to all elements of an affine vector space $\mathcal{V} \oplus c$ of dimension $\geq d + 1$ for arbitrary constant c is zero, that is $\bigoplus_{v \in \mathcal{V} \oplus c} v = \bigoplus_{v \in \mathcal{V} \oplus c} f(v) = 0$. This property is exploited by higher-order differential attacks [Knu94]. Here we focus on the security of HADESMIMC against such an attack when instantiated over $\mathbb{F}_p$.

$\mathbb{F}_{2^n}$ versus $\mathbb{F}_p$. First of all, let us emphasize an important difference between the higher-order differential attack on $\mathbb{F}_{2^n}$ and on $\mathbb{F}_p$. As we just recalled, given a function $f(\cdot)$ of degree d, then $\bigoplus_{v\in\mathcal{V}\oplus c} f(v) = 0$ for each vector space $\mathcal{V}$ of dimension $\geq d+1$ is zero. The crucial point here is that *the previous result holds if $\mathcal{V}$ is a (sub)space, and not only a generic set of elements.*

While $\mathbb{F}_{2^m}$ is always a subspace of $\mathbb{F}_{2^n}$ for each $m \leq n$, the only subspaces of $\mathbb{F}_p$ are $\{0\}$ and $\mathbb{F}_p$. It follows that the biggest subspace of $(\mathbb{F}_p)^t$ has dimension t, with respect to the biggest subspace of $(\mathbb{F}_{2^n})^t$ which has dimension $n \cdot t = N$. As a result, in the case in which a cipher is instantiated over $\mathbb{F}_p$, a lower degree (and hence a smaller number of rounds) is sufficient to protect it against the higher-order differential attack w.r.t. the number of rounds for the $\mathbb{F}_{2^n}$ case.

Security Analysis: HadesMiMC instantiated over $\mathbb{F}_p$. Due to the discussion just given, the biggest (non-trivial) subspace of $(\mathbb{F}_p)^t$ has dimension at most $t-1$. Thus, HadesMiMC is secure against higher-order differential attacks if its degree is bigger than $t-1$. It follows that if HadesMiMC (instantiated over $\mathbb{F}_p$) is secure against the interpolation attack, then it is also secure against the higher-order differential attack. In other words, the number of rounds necessary to guarantee the security of HadesMiMC against the interpolation attack is sufficient to guarantee the security against the higher-order differential one as well.

12.3.4. Low-Data Scenario (HadesMiMC instantiated over $\mathbb{F}_{2^n}$)

Finally, for the signature application (Picnic), we provide a security analysis of HadesMiMC instantiated over $\mathbb{F}_{2^n}$ in the "low-data scenario" only, that is, the case in which the attacker can only use the knowledge of a *single* (plaintext, ciphertext) pair to set up the attack.

To derive the sufficient number of rounds to guarantee security, we consider two attacks, that is, *(1st)* Meet-in-the-Middle attacks and *(2nd)* the GCD attack and its generalization, the Gröbner basis attack. Note that any other attack – even the (truncated) differential one with high probability (e.g. prob. 1) – requires at least 2 pairs of texts, hence it is not applicable in this scenario.

Since differential attacks do not apply in this scenario, we consider a more "aggressive" version of our design HadesMiMC. In particular, we replace the assumption that the MixLayer is an MDS matrix with the "weaker" request that it is invertible[27], together with the property that $M^i = \prod_{j=1}^{i} M$ has no zero coefficient[28] for each $1 \leq i \leq R$ (where R is the number of rounds). For an efficient implementation, we also require that at least one $(t-1)\times(t-1)$ submatrix of M is invertible (a concrete example is given in Sect. 12.5.2). Moreover, also the assumption that "$R_F = 2R_f$ must be even" can be relaxed, that is, we do not force the cipher to be "symmetric"[29] (in other words, R_F can be odd).

GCD and Gröbner Basis Attack. About the GCD and Gröbner basis attacks, due to the analysis already given, the number of rounds $R_F \geq 2$ and R_P must satisfy the conditions $R_P + R_F \geq 4 + \lceil n \cdot \log_3 2 \rceil - \lfloor 2 \cdot \log_3(n) \rfloor$, $R_P + t \cdot R_F \geq R^{2nd-Grob}(N,t)$ and $R_F \geq R^{3rd-Grob}(N,t,R_P)$ respectively for the two attacks in order to guarantee security.

MitM (statistical) Attack. About the MitM attack in the low-data scenario, we take into account the similarity between HadesMiMC and AES, and we mention the existence of low-data key-recovery attacks up to 4-round AES [BDD+12] that require only 1 (plaintext, ciphertext) pair. Such MitM attacks exploit the low diffusion of the AES key schedule and the fact that two rounds of

[27] Give A and B invertible lower and upper triangular matrices, $M = B \times A$ is invertible.

[28] If no matrix satisfies this condition, then the idea is to choose a matrix that minimizes the total number of zero coefficients.

[29] If $R_F = 2R_f + 1$, the distribution of the rounds is: $\big[R_f + 1$ rounds with full S-Box layer$\big]$ + $\big[R_P$ rounds with partial S-Box layer$\big]$ + $\big[R_f$ rounds with full S-Box layer$\big]$.

AES are necessary to provide full diffusion. Since both one round of our key schedule and one round of HADESMIMC provide maximum diffusion, we conjecture that our design with $R_F = 3$ rounds with a full S-Box layer is protected from this kind of attack.

Why $R_F = 2$ rounds with full S-Box layers are not sufficient to guarantee security against MitM? Working in $\mathbb{F}_{2^n}$ (similar for $\mathbb{F}_p$), consider the linear layer proposed in (12.14). If $R_F = 2$ and $R_P = 0$, a MitM attack can be potentially set up. Indeed, given a plaintext p and the corresponding ciphertext c, note that it is possible to construct several equivalences of the form

$$3 \times \left[\text{S-Box}(p_j \oplus k_j^{(0)}) \oplus \text{S-Box}(p_l \oplus k_l^{(0)})\right] \oplus k_j^{(1)} \oplus k_l^{(1)} = \text{S-Box}^{-1}(c_j \oplus k_j^{(2)}) \oplus \text{S-Box}^{-1}(c_l \oplus k_l^{(2)})$$

for each $0 \leq j, l < t$, where $3 \equiv \text{0x03}$ and where the final MixLayer has been omitted. These can be used to speed up a brute-force attack. A similar result can be obtained for small $R_P \geq 1$.

Three rounds with a full S-Box layer allow to prevent such an attack. Indeed, in this case, the equations take the form:

$$\begin{aligned} 3 \times \Big\{ &\text{S-Box}\Big[k_j^{(1)} \oplus \bigoplus_{i=0}^{t-1} M_{j,i} \cdot \text{S-Box}\big(p_i \oplus k_i^{(0)}\big)\Big] \oplus \text{S-Box}\Big[k_l^{(1)} \oplus \bigoplus_{i=0}^{t-1} M_{l,i} \cdot \text{S-Box}\big(p_i \oplus k_i^{(0)}\big)\Big]\Big\} \\ &\oplus k_j^{(2)} \oplus k_l^{(2)} = \text{S-Box}^{-1}(c_j \oplus k_j^{(3)}) \oplus \text{S-Box}^{-1}(c_l \oplus k_l^{(3)}) \end{aligned}$$

for each $0 \leq j, l < t$, where as before $3 \equiv \text{0x03}$ and where the final MixLayer has been omitted. Since all words of $k^{(0)}$ are involved as inputs of different S-Boxes, we argue that it is (rather) hard to exploit such equivalences to speed up the brute-force attack.

Conclusion. In the low-data scenario, we claim that $R = R_F + R_P$ rounds of HADESMIMC instantiated by $\text{S-Box}(x) = x^3$ - both for $GF(p)$ and $GF(2^n)$ - are sufficient to provide security if $R_F \geq 3$ and

$$\begin{cases} R_P + R_F \geq 4 + \lceil n \cdot \log_3 2 \rceil - \lfloor 2 \cdot \log_3(n) \rfloor \\ R_P + t \cdot R_F \geq \lceil N/[2 \cdot \log_2((2p-1)/3)] \rceil + \lceil N/[2 \cdot \log_2(27/4)] \rceil \\ R_F \geq R^{3rd-Grob}(N, t, R_P) \equiv 2 + \log_3(2) \cdot \left(\frac{N}{2t+R_P} + 2 \cdot \log_2(2t + R_P) - 2 \cdot \log_2(2t) \right) \end{cases}$$

12.4. Number of Rounds Needed for Security

The design goal of HADESMIMC is to offer a cipher optimized for schemes whose performance critically depends on the MULTdepth/ANDdepth, the number of MULTs/ANDs, or the number of MULTs/ANDs per bit. We thus try to be as close to the number of rounds needed for security as possible.

Besides the possibility to choose the size of the S-Box, one of the strengths of our design is the freedom to choose the ratio between the number of rounds R_F with full S-Box layer and the number of rounds R_P with partial S-Box layer. For the applications that we have in mind, here we limit ourselves to optimize HADESMIMC w.r.t. two different metrics:

Number of Multiplications/S-Box: this metric is the best one in order to describe the cost in the case of MPC applications. Motivated by real-life applications, our goal is to reduce the total runtime (as we describe in the following). Since the main bottleneck of a protocol run on top of the SPDZ framework is the triple generation mechanism, which is given by the number of multiplications/non-linear operations, the goal is to minimize the total number of multiplications (which is proportional to the number of S-Boxes);

Number of Multiplications/S-Box × Field Size: this metric is the best one in order to describe the cost in the case of Picnic Post-Quantum Signature scheme (see [CDG+17] for details).

We recall that in the first case HadesMiMC is instantiated over $\mathbb{F}_p$, while in the second case it is instantiated over $\mathbb{F}_{2^n}$ (in the "low-data scenario" only).

Preliminary. HadesMiMC is secure if and only if at least one of the following two systems of inequalities is satisfied:

$$R_F \geq \max\{R_F^{stat}(N,t), R_F^{Grobner}(N,t)\} \qquad \text{and} \qquad R_P + R_F \geq R^{inter}(N,t)$$

where $R^{inter}(N,t)$ is defined in (12.4), $R_F^{Grober}(N,t)$ in (12.7) and R_F^{stat} in (12.3), *or*

$$\begin{cases} R_F \geq \max\big\{R_F^{stat}; R^{3rd-Grob}(N,t,R_P)\big\} \\ R_P + R_F \geq \Psi^{(1)}(N,t) \equiv \max\big\{R^{inter}(N,t); R^{1st-Grob}(N,t)\big\} \\ R_P + t \cdot R_F \geq \Psi^{(t)}(N,t) \equiv R^{2nd-Grob}(N,t) \end{cases}$$

where $R^{1st-Grob}(N,t), R^{2nd-Grob}(N,t), R^{3rd-Grob}(N,t)$ are defined as in (12.8).

In other words, HadesMiMC results secure if – *for every attack* – (at least) one of the following inequality is satisfied

$$R_F \geq \Phi^F(N,t) \quad \text{or} \quad R_P + \varphi(t)\cdot R_F \geq \Phi^P(N,t) \quad \text{or} \quad R_F \geq \Phi(N,t,R_P)$$

where $\Phi(N,t,R_P), \Phi^F(N,t), \Phi^P(N,t)$ and $\varphi(t)$ are functions that depend on the attack, where $\varphi(t) = 1$ or $\varphi(t) = t$.

Note that – in our design strategy – we always exploit the "Wide-Trail" strategy in order to guarantee security against statistical attacks. In other words, for this class of attacks, we only work with rounds with full S-Box layer in order to guarantee security. HadesMiMC results secure against statistical attacks if

$$R_F^{stat} \geq \Phi^{stat}(N,t)$$

where $\Phi^{stat}(N,t)$ – provided in (12.3) – is equal to 6 or 8. Thus, given

$$R_F = R_F^{stat} + R'_F \geq R_F^{stat},$$

we are actually looking for *the best ratio between R'_F and R_P that minimizes the total number of S-Boxes.*

12.4.1. Minimize "Number of S-Boxes" – HadesMiMC over $\mathbb{F}_p$

Here we mainly focus on minimizing the number of S-Boxes, since this is the metric that best describes the cost of the applications that we have in mind. In other words, for given n and t, the goal is to find the best ratio between R_P and R_F that minimizes the total number of S-Boxes, given by

$$\textit{minimum number of S-Boxes } = t \cdot R_F + R_P \tag{12.9}$$

where $t \geq 2$ and where the number of non-linear operations is proportional to the number of S-Boxes.

As supplementary material, we provide a script that given an input N, returns the best t and the best ratio between R_P and R_F that minimizes the cost metric (in this case, the total number of S-Boxes). *For each possible value of t* where $2 \leq t \leq \frac{N}{\log_2\{[(2N)/(\log_2(N+1)+1)]+1\}}$ (where this upper bound guarantees the existence of an MDS matrix), the script finds the best ratio between R_P and R_F that minimize the number of S-Boxes for that particular t. Then, it simply finds the best value of t that minimizes the total number of non-linear operations.

For our MPC applications, we also provide a variant of such script which take in input both N and t and returns the best ratio between R_P and R_F that minimizes the cost metric for that particular input.

What is the Best Ratio between $\mathbf{R_F}$ and $\mathbf{R_P}$? Details for a Simplified Case. In order to understand what a certain ratio between R_F and R_P is the best one, here we study in details a possible way to find – without brute-force – the best ratio between R_F and R_P which minimizes the cost metric (12.9) and which guarantees security *for a simplified case, i.e. assuming* HadesMiMC *is secure if the following inequalities are satisfied*[30]

$$\begin{cases} R_F \geq R_F^{stat} \qquad \text{and} \qquad R_P \geq 0; \\ R_P + R_F \geq \Psi^{(1)}(N,t) \equiv \max\{R^{inter}(N,t); R^{1st-Grob}(N,t)\} \\ R_P + t \cdot R_F \geq \Psi^{(t)}(N,t) \equiv R^{2nd-Grob}(N,t) \end{cases}$$

The goal is to find the best ratio between R'_F (where $R_F = R_F^{stat} + R'_F \geq R_F^{stat}$) and R_P that minimizes the total number of S-Boxes, where both $\Psi^{(1)}(N,t)$ and $\Psi^{(t)}(N,t)$ are fixed (since N and t are fixed).

First Case. Firstly, we analyze the case in which

$$\Psi^{(t)} < t \times \Psi^{(1)} \qquad \text{and} \qquad \Psi^{(t)} \leq \Psi^{(1)} + R_F^{stat} \times (t-1) \tag{12.10}$$

which corresponds to the case in which, if the second inequality $R_P + R_F \geq \Psi^{(1)}(N,t)$ is satisfied, then the third one $R_P + t \cdot R_F \geq \Psi^{(t)}(N,t)$ is also satisfied:

$$R_P + t \cdot R_F \Big|_{R_P + R_F \geq \Psi^{(1)}(N,t)} \geq \Psi^{(1)} + R_F \cdot (t-1) \geq \Psi^{(1)} + R_F^{stat} \cdot (t-1) \geq \Psi^{(t)}.$$

Thus, under the assumption (12.10), we can just focus on $R_P + R_F \geq \Psi^{(1)}(N,t)$. It follows that by combining eq. (12.9) (i.e., number of S-Boxes) and eq. $R_P + R_F \geq \Psi^{(1)}(N,t)$, the cost is upper bounded by

$$t \cdot R_F + R_P \Big|_{R_P + R_F \geq \Psi^{(1)}} \leq R_F(t-1) + \Psi^{(1)}$$

which is minimized by taking the minimum value of R_F (where note that $\Psi^{(1)}$ is fixed for t and N fixed). As a result, to minimize the total number of rounds, the idea is to minimize the number of rounds R_F with full S-Box layer, that is

$$R_F = R_F^{stat} \quad \text{and} \quad R_P \geq \max\left\{0; R^{inter} - R_F; R^{1st-Grob}(N,t) - R_F\right\}.$$

Second Case. Secondly, we analyze the case in which

$$\Psi^{(t)} \geq t \times \Psi^{(1)} \qquad \text{and} \qquad \Psi^{(t)} > \Psi^{(1)} + R_F^{stat} \times (t-1) \tag{12.11}$$

which corresponds to the case in which, if the third inequality $R_P + t \cdot R_F \geq \Psi^{(t)}(N,t)$ is satisfied, then the second one $R_P + R_F \geq \Psi^{(1)}(N,t)$ is also satisfied.

Thus, under the assumption (12.11), let us limit to consider the condition $R_P + t \cdot R_F \geq \Psi^{(t)}(N,t)$. Since the number $R_P + t \cdot R_F$ corresponds to the total number of S-Boxes, it follows that each choice

[30] This corresponds to the case in which we guarantee security against Gröbner basis attack by using rounds with partial S-Box layer, and in which we *assume that inequality $R_F \geq R^{3rd-Grob}(N,t,R_P)$ from (12.8) is always satisfied.*

of R_P and R_F that satisfy $R_P + t \cdot R_F = \Psi^{(t)}(N,t)$ – where[31] $R_F^{stat} \le R_F \le \frac{\Psi^{(t)}}{t}$ – also minimizes the number of S-Boxes.

Thus, *which choice of R_F and R_P is the best one?* If one focuses only on non-linear operations (that is, linear operations are cost-less), then there is no reason to choice any particular R_F. However, in general this is not the case. Even if the cost of linear operations is negligible compares to the cost of non-linear ones, note that t rounds with partial S-Box layer counts more linear operations that the a single round with full S-Box layer. In particular, the number of linear operations are t^2 for each round with full S-Box layer and $t \times (3t-2) = 3t^2 - 2t$ for t rounds with partial S-Box layer. As a result, if one would minimize also the number of linear operations, the correct metric to use is

$$t \cdot R_F + \big[1 + \varepsilon(N,t)\big] \cdot R_P$$

where $\varepsilon(N,t) \ge 0$ represents the cost of the linear operations[32] (where $\varepsilon(N,t) = 0$ if one does not care of the cost of linear operations).

In such a case, under the assumption (12.11), the total cost is given by

$$t \cdot R_F + \big[1 + \varepsilon(N,t)\big] \cdot R_P \Bigg|_{R_P + t\cdot R_F \ge \Psi^{(t)}(N,t)} \ge \big[1 + \varepsilon(N,t)\big] \cdot \Psi^{(t)}(N,t) - \varepsilon(N,t) \cdot t \cdot R_F$$

which is minimized by taking the maximum (possible) number of rounds with full S-Box layer. As a result, under the assumption (12.11), the best choice is

$$R_F = \max\left\{R_F^{stat}; 2 \times \left\lfloor \frac{R^{2nd-Grob}(N,t)}{2t} \right\rfloor\right\}, \; R_P = \max\left\{0; \left\lceil R^{2nd-Grob} \right\rceil - t \cdot R_F\right\}$$

Third Case. Finally, we study the case in which

$$\Psi^{(t)} < t \times \Psi^{(1)} \qquad \text{and} \qquad \Psi^{(t)} > \Psi^{(1)} + R_F^{stat} \times (t-1). \tag{12.12}$$

In this case, note that:

- the case $R_F^{stat} \le R_F \le \frac{\Psi^{(t)} - \Psi^{(1)}}{t-1}$ corresponds to the case in which, if the third inequality $R_P + t \cdot R_F \ge \Psi^{(t)}(N,t)$ is satisfied, then the second one $R_P + R_F \ge \Psi^{(1)}(N,t)$ is also satisfied;
- the case $\frac{\Psi^{(t)} - \Psi^{(1)}}{t-1} \le R_F \le \Psi^{(1)}$ corresponds to the case in which, if the second inequality is satisfied, then the third one is also satisfied.

Re-using the results given before it makes sense to maximize R_F when the third inequality predominates on the second one, and to minimize R_F when the second inequality predominates on the third one. As a result, under the assumption (12.12), the best choice is given by:

$$R_F = \max\left\{R_F^{stat}; 2 \cdot \left\lfloor \frac{\Psi^{(t)} - \Psi^{(1)}}{2(t-1)} \right\rfloor\right\} \quad \text{and} \quad R_P = \max\left\{0; R^{2nd-Grob}(N,t) - (t \times R_F)\right\}$$

where $\Psi^{(t)}(N,t) - \Psi^{(1)}(N,t) = R^{2nd-Grob}(N,t) - \max\big\{R^{inter}(N,t); R^{1st-Grob}(N,t)\big\}$.

[31] The upper bound is due to the fact that both $R_P \ge 0$ and $R_P + t \cdot R_F \ge \Psi^{(t)}(N,t)$ must be satisfied.

[32] E.g. assume that the cost of one non-linear operation in $\mathbb{F}_p$ – where $\log_2(p) \approx n$ – is equal to the cost of $n \cdot \log(n)$ linear operations. It follows that $\varepsilon(N,t)$ is well approximated by $1 + \frac{t-2}{t + n\log(n)}$.

12.4.2. Minimize "Number of S-Boxes × Field Size"

Secondly, we consider the metric given by "number of S-Boxes × field size", which well describes the cost of the Picnic PQ-Signature Scheme – where HadesMiMC instantiated over $\mathbb{F}_{2^n}$ (low-data case). In this case, for each N and t, the goal is to find the best ratio of R_P and R'_F (where $R_F = R_F^{stat} + R'_F \geq R_F$) for which the following cost is minimized

$$n \times \big(t \cdot R_F + R_P\big) = N \cdot R_F + n \cdot R_P. \tag{12.13}$$

If both n and t are fixed, this metric is proportional to the one given before (that is, it is equal to the one given in (12.9) times a factor n). Thus, the results given in the previous section hold also for this metric. As before, for the case in which t is not fixed, we provide *a script that takes in input N and returns the best t and the best ratio between R_F and R_P that minimizes the metric given in (12.9).*

12.4.3. Concrete Instantiations of HadesMiMC

Based on the security analysis just proposed, in Tables 12.2 we present concrete instantiations of HadesMiMC for different security level and/or applications.

Table 12.2.: *A range of different parameter sets for* HadesMiMC*– instantiated by S-Box$(x) = x^3$ – offering different trade-offs.* The first set is for AES-like security parameter (128 bit). The second and the third sets are resp. for MPC and Post-Quantum Signature applications (low-data scenario only). Finally, the last set includes an example of toy version useful to facilitate third-party cryptanalysis.

Text Size $N = n \times t$	**S-Box Size** (n or $\log_2 p$)	**# S-Boxes** (t)	**Rounds R_F** (Full S-Box)	**Rounds R_P** (Partial S-Box)	$(\mathbb{F}_p)^t$	$(\mathbb{F}_{2^n})^t$ **Low-Data**
128	8	16	8	4	✓	
128	16	8	6	10	✓	
256	128	2	10	75	✓	
512	128	4	12	74	✓	
1 024	128	8	14	76	✓	
2 048	128	16	18	86	✓	
4 096	128	32	20	106	✓	
≈ 256	3	86	3	1		✓
32	8	4	6	5	✓	

Reduced and Toy Versions. Many classes of cryptanalytic attacks become more difficult with an increased number of rounds. In order to facilitate third-party cryptanalysis and estimate the security margin, reduced-round variants need to be considered. Hence we encourage to study round-reduced variants of HadesMiMC where the symmetry around the middle is kept. Moreover, to encourage cryptanalysis of HadesMiMC, we remark that it is possible to specify toy versions of our cipher which aim at achieving e.g. 32 or 64 bits of security.

12.5. MPC and Post-Quantum Signature Applications

Remark. *Since I did not work on the practical applications/implementations of MiMC, I limit myself to recall here the main results and I refer to [GLR+19] for a detailed discussion on it. The results of this section are due to the work of Dragos Rotaru (MPC applications) and Sebastian Ramacher and Markus Schofnegger (PQ-Signature application) respectively.*

Table 12.3.: Two-party costs for MiMC and HadesMiMC over a LAN network. *Here we limit ourselves to give the results just for $t = 2, 4$ and 32, and we refer to [GLR+19, App. F] for the corresponding results for $t = 8$ and 16.*

Mode$_t$	**Online cost**				*Preproc per block*	
	(MPC) Rounds	Openings	Best Lat (ms/$\mathbb{F}_p$)	Throughput ($\mathbb{F}_p$/s)	ms	MBytes
HadesMiMC$_2$	84	273	3.24	11796	3.06	0.17
MiMC$_2$	73	438	2.98	9681	4.86	0.27
HadesMiMC$_4$	84	342	1.64	16284	1.9	0.10
MiMC$_4$	73	876	1.52	10856	4.86	0.27
HadesMiMC$_{32}$	116	2394	0.36	3972	1.66	0.09
MiMC$_{32}$	73	7008	0.30	11021	4.86	0.27

12.5.1. MPC Experiments

For MPC applications, we evaluated HadesMiMC cipher using the SPDZ framework [KSS13] within a prime field $\mathbb{F}_p$. To briefly understand the MPC details, we denote by $[x]$ a sharing of x, where each party P_i holds a random $x_i \in \mathbb{F}_p$. The process of parties reconstructing x is called an opening, i.e., going from a shared value $[x]$ to a public value x known to all parties.

As with most MPC frameworks, a protocol is split into two steps: an input-independent preprocessing phase where parties generate random Beaver triples $[a] = [b] \cdot [c]$, and an input-dependent online phase where parties share their inputs and use the triples generated in the preprocessing phase. The cost of a multiplication between two secret values $[z] \leftarrow [x] \cdot [y]$ is twofold: one Beaver triple generated in the preprocessing phase as well as two openings and one round of communications in the online phase. Since secret shared multiplications can be done in parallel, the number of communication rounds in the online phase is given by the multiplicative depth of the circuit to be evaluated. Linear operations such as additions and multiplications by public scalars are non-interactive and require only a small computational overhead.

In our setting, both the key $[k]$ and the message $[m]$ are shared between the parties. In the following, we will show how to compute the S-Box (note that everything else is linear and can be computed locally). The trivial way to compute the cubing is by computing $[x^2] \leftarrow [x] \cdot [x]$ and then $[x^3] \leftarrow [x^2] \cdot [x]$. This can be done with two communication rounds and it has an online cost of 3 openings and uses two triples. We use the Grassi *et al.* version [GRR+16] to reduce the online cost to one communication round with the same amount of openings and triples.

Standard Benchmarks. We implemented and benchmarked HadesMiMC in SPDZ between two computers with commodity hardware connected via a 1 GB/s LAN connection with a round-trip time of 0.3 ms. Here, latency represents the best running time of a single cipher evaluation, whereas throughput means the maximum number of field elements that can be encrypted in parallel per second. In Table 12.3 there are the two extremes: HadesMiMC encrypting messages using $t = 2$ blocks with $n = 128$ bits per block (32 bytes) as well as $t = 32$ blocks with $n = 128$ bits per block (512 bytes).

Amortizing Preprocessing Cost. This can be achieved if one chooses to encrypt a large number of blocks at once. Encrypting $2 \cdot 128$ blocks (32 bytes) requires 0.17 MBytes of data per block with HadesMiMC. The preprocessing cost per block decreases and at $t = 32$ with HadesMiMC reaches 0.09 MB per block. This is in contrast with MiMC which keeps the preprocessing cost constant at 0.27 MB per block (see [GLR+19, App. F] for details). As a consequence, the total runtime of HadesMiMC (see [GLR+19, App. F] for details) is smaller than MiMC since the main bottleneck for MPC with dishonest majority is the triple generation.

12.5.2. Post-Quantum Signatures from Symmetric-Key Primitives

Finally, we analyze HadesMiMC as a replacement of LowMC in Picnic. The advent of efficient zero-knowledge proof systems for arithmetic circuits [GMO16; CDG+17; AHIV17; KKW18] enable an alternative design paradigm to hash-based signatures for constructing post-quantum secure signature schemes built from symmetric-key primitives. Following this design strategy, Picnic [CDG+17] using LowMC as an underlying symmetric-key primitive was submitted as part of the NIST PQC effort. Besides signature schemes, this paradigm has also been used to construct privacy-preserving variants, such as ring and variants of group signatures [DRS18b; BEF18; KKW18] and double-authenticating preventing signatures [DRS18a].

Regardless of the underlying proof system, the involved proof sizes are mainly influenced by the number of multiplication gates and the field size. For ZKB++ and KKW, the proof sizes are linear in the number of multiplication gates. Consequently, for selecting the optimal symmetric-key primitive, the product of the number of multiplication gates and the field size is the most important metric when optimizing for signature size.

For the evaluation of HadesMiMC in this context, we focus on Picnic and its underlying proof system, ZKB++. In Picnic, signatures consist of a zero-knowledge proof of knowledge of a preimage x to a one-way function f, where the image $y = f(x)$ is the public key and f is instantiated using LowMC. Interestingly, a potential attacker is only able to obtain a single (plaintext, ciphertext) pair, thus selecting a LowMC parameterization for a low-data scenario is sufficient. When instantiating the signature scheme with HadesMiMC instead, we can also make use of this fact and derive round numbers appropriate for the low-data scenario. Additionally, this allows us to choose invertible matrices for the key schedule and for the linear layer which do not necessarily need to be MDS matrices. For example, by choosing a $t \times t$ matrix M with elements in $\mathbb{F}_p$, where t denotes the number of words and

$$M_{i,j} = \begin{cases} 2 & \text{if } i = j, \\ 1 & \text{otherwise,} \end{cases} \tag{12.14}$$

the sum to assign to each of the words in the linear layer differs in only one term. Most of this sum can thus be precomputed in each layer, and the additional term is added separately for each word. Using this method, the expensive matrix-vector multiplication can be avoided and the computational effort is reduced from $\mathcal{O}(t^2)$ operations to $\mathcal{O}(t)$ operations. Using the script presented before, it turns out that for $N = 256$ the metric “number of multiplications × field size” is minimized by $n = 3$ (so $t = 86$) and by $R_F = 3$ (so $R_P = 1$).

Experiments and Results. When using HadesMiMC, we can observe much better results than with MiMC. The new design strategy helps to reduce both the computational cost and the proof size significantly, as Table 12.4 demonstrates for selected instances of HadesMiMC and MiMC. Due to faster computations and significantly smaller proof sizes, only instantiations using binary fields are listed for HadesMiMC in this comparison. On the other hand, instances using prime fields are chosen for MiMC, because they appear to provide a good trade-off between the signature size and the computation time. We also note that in the table we focus on the time spent to compute shared circuits, and omit the overhead independent of the cipher.

In comparison with LowMC-based instantiations, we are able to obtain smaller signatures using HadesMiMC – even when compared to the smallest possible LowMC-based signature with only 1 S-Box and 363 rounds totaling a view size of 1089 bits. The view size of HadesMiMC-$(3, 86)$ totals at 777 bits, which amounts to an improvement by a factor of 1.40. Performance-wise, the

Table 12.4.: Computation times of circuits and view sizes in ZKB++ using HadesMiMC and MiMC, where $N = n \cdot t \approx 256$, and LowMC-$(N, m, r)$ with block-/key-size N, m S-Boxes and r rounds. The 272-bit prime number used in MiMC allows for faster computations than the 256-bit one.

Scheme	Proof Generation	Verification	View Size
HadesMiMC-$(3, 86)$	**0.40 ms**	**0.29 ms**	**777 bits**
HadesMiMC*-$(3, 86)$	0.49 ms	0.30 ms	1032 bits
LowMC-$(256, 10, 38)$	3.74 ms	3.52 ms	1140 bits
LowMC-$(256, 1, 363)$	9.55 ms	7.12 ms	1089 bits
MiMC-$(256, 1)$	303.58 ms	161.65 ms	83456 bits
MiMC-$(272, 1)$	90.84 ms	47.18 ms	94112 bits

3-bit instance results in much faster signing and verification times than LowMC thanks to a highly parallelized implementation, even when considering a LowMC implementation with recent optimizations [DKP+19]. For completeness, the table includes the case HadesMiMC* with $R_F = 4$ and $R_P = 0$ which provides more security margin (with respect to MitM attacks) than the version $R_F = 3$ and $R_P = 1$. Also in this case, our design is more competitive than LowMC.

13 Open Problems – MiMC and its Generalizations

- Our current cryptanalysis results suggest that almost all attacks work (approximately) in the same way when the ciphers GMiMC and HadesMiMC are instantiated in $\mathbb{F}_{2^n}$ or in $\mathbb{F}_p$. The only exception is given by the higher-order differential attack, for which two different scenarios arise. As we have just seen, given a function $f(\cdot)$ of degree d, it is possible to prove that $\bigoplus_{x \in V \oplus a} f(x) = 0$ for each subspace V such that $\deg(f) + 1 \leq \dim(V)$. In the case of $\mathbb{F}_{2^n}$, every set $\mathbb{F}_{2^m}$ for $m \leq n$ is a subspace of $\mathbb{F}_{2^n}$. In other words, the dimension of the biggest subspace of $\mathbb{F}_{2^n}$ is n. Instead, if one considers the case $\mathbb{F}_p$, the only subspaces of $\mathbb{F}_p$ are $\{0\}$ and $\mathbb{F}_p$. This means that the biggest subspace of $\mathbb{F}_p$ has dimension 1. As a result, higher-order differential attack seems to be (much) more competitive in $\mathbb{F}_{2^n}$ than in $\mathbb{F}_p$.

  The problem to fill this gap – by studying competitive higher-order differential attacks on $\mathbb{F}_p$ – is left as future problem.

- One of the major problem that we faced while studying the security of GMiMC is that – to the best of our knowledge – almost no result about the application of the higher-order differential attack and/or of the division property on Generalized Feistel Network is present in the literature. As future open problem, it would be important to derive a formula about the growth of the algebraic degree for Generalized Feistel Network, similar to the one proposed in [BCC11] for SPN ciphers.

- For our target applications, we decided to apply the Hades Strategy only to SHARK [DKR97]. It could be interesting to apply the same strategy also to other ciphers, like AES. Could such strategy give any advantage also for masking or other applications?

- In [CDK+18], authors study the provable security of *linear* SPN designs. A SPN cipher $E_k(\cdot) : \mathbb{F}_{2^N} \to \mathbb{F}_{2^N}$ where $N = n \cdot t$ is "linear" if the non-linear operation of each round transformation $R(\cdot) : \mathbb{F}_{2^N} \to \mathbb{F}_{2^N}$ is defined as a concatenation of $t \geq 2$ *independent* non-linear transformations $S_1(\cdot) \| S_2(\cdot) \| ... \| S_t(\cdot)$, that is

  $$R(\cdot) = L \circ \big[S_1(\cdot) \| S_2(\cdot) \| ... \| S_t(\cdot)\big]$$

  where $S_i(\cdot) : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ for each $i = 1, ..., t$ and where $L : \mathbb{F}_{2^N} \to \mathbb{F}_{2^N}$ is a linear operation. In that paper, authors show that 3 SPN rounds are necessary and sufficient for security (if common assumption on the keys and on the details of the round transformation are satisfied).

  While similar analysis of Feistel designs are already present in the literature (see e.g. [LR88] and [Pat03; Pat04]), to the best of our knowledge no-one considers the security of Partial-SPN cipher or of ciphers based on the innovative Hades strategy from the provable security point of view. This is left as an open problem for future work.

References

[AÅBL12] M. A. Abdelraheem, M. Ågren, P. Beelen, and G. Leander. On the Distribution of Linear Biases: Three Instructive Examples. In: Advances in Cryptology – CRYPTO 2012. Ed. by R. Safavi-Naini and R. Canetti. Vol. 7417. LNCS. Springer, 2012, pp. 50–67. DOI: 10.1007/978-3-642-32009-5_4 (p. 203).

[ADK+14] M. R. Albrecht, B. Driessen, E. B. Kavun, G. Leander, C. Paar, and T. Yalçin. Block Ciphers - Focus on the Linear Layer (feat. PRIDE). In: Advances in Cryptology – CRYPTO 2014. Ed. by J. A. Garay and R. Gennaro. Vol. 8616. LNCS. Springer, 2014, pp. 57–76. DOI: 10.1007/978-3-662-44371-2_4 (pp. 98, 99).

[AGP+18] M. R. Albrecht, L. Grassi, L. Perrin, S. Ramacher, C. Rechberger, D. Rotaru, A. Roy, and M. Schofnegger. Feistel Structures for MPC, and more. In Submission. 2018 (pp. 6, 197, 211–214, 228, 234, 235).

[AGR+16] M. R. Albrecht, L. Grassi, C. Rechberger, A. Roy, and T. Tiessen. MiMC: Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity. In: Advances in Cryptology – ASIACRYPT 2016. Ed. by J. H. Cheon and T. Takagi. Vol. 10031. LNCS. Springer, 2016, pp. 191–219. DOI: 10.1007/978-3-662-53887-6_7 (pp. 197–199, 205, 207, 211, 217, 234, 237, 238, 244, 255).

[AHIV17] S. Ames, C. Hazay, Y. Ishai, and M. Venkitasubramaniam. Ligero: Lightweight Sublinear Arguments Without a Trusted Setup. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security – CCS 2017. 2017, pp. 2087–2104. DOI: 10.1145/3133956.3134104 (pp. 17, 268).

[AM] J.-P. Aumasson and W. Meier. Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi. presented at the Rump Session of Cryptographic Hardware and Embedded Systems - CHES 2009. URL: https://131002.net/data/papers/AM09.pdf (p. 224).

[ARS+15] M. R. Albrecht, C. Rechberger, T. Schneider, T. Tiessen, and M. Zohner. Ciphers for MPC and FHE. In: Advances in Cryptology – EUROCRYPT 2015. Ed. by E. Oswald and M. Fischlin. Vol. 9056. LNCS. Springer, 2015, pp. 430–454. DOI: 10.1007/978-3-662-46800-5_17 (pp. 198, 237, 238, 240, 242).

[BA08] B. Bahrak and M. R. Aref. Impossible differential attack on seven-round AES-128. In: IET Information Security 2.2 (2008), pp. 28–32. DOI: 10.1049/iet-ifs:20070078 (p. 38).

[BB02] E. Barkan and E. Biham. In How Many Ways Can You Write Rijndael? In: Advances in Cryptology – ASIACRYPT 2002. Ed. by Y. Zheng. Vol. 2501. LNCS. Springer, 2002, pp. 160–175. DOI: 10.1007/3-540-36178-2_10 (p. 47).

[BBBF18] D. Boneh, J. Bonneau, B. Bünz, and B. Fisch. Verifiable Delay Functions. In: Advances in Cryptology – CRYPTO 2018. Ed. by H. Shacham and A. Boldyreva. Vol. 10991. LNCS. Springer, 2018, pp. 757–788. DOI: 10.1007/978-3-319-96884-1_25 (p. 210).

[BBD+98] E. Biham, A. Biryukov, O. Dunkelman, E. Richardson, and A. Shamir. Initial Observations on Skipjack: Cryptanalysis of Skipjack-3XOR. In: Selected Areas in Cryptography - SAC 1998. Ed. by S. E. Tavares and H. Meijer. Vol. 1556. LNCS. Springer, 1998, pp. 362–376. DOI: 10.1007/3-540-48892-8_27 (pp. 44, 108).

[BBHR18] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev. Scalable, transparent, and post-quantum secure computational integrity. Cryptology ePrint Archive, Report 2018/046. https://eprint.iacr.org/2018/046. 2018 (p. 210).

[BBI+15] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, and F. Regazzoni. Midori: A Block Cipher for Low Energy. In: Advances in Cryptology – ASIACRYPT 2015. Ed. by T. Iwata and J. H. Cheon. Vol. 9453. LNCS. Springer, 2015, pp. 411–436. DOI: 10.1007/978-3-662-48800-3_17 (pp. 98, 99).

[BBK14] A. Biryukov, C. Bouillaguet, and D. Khovratovich. Cryptographic Schemes Based on the ASASA Structure: Black-Box, White-Box, and Public-Key (Extended Abstract). In: Advances in Cryptology – ASIACRYPT 2014. Ed. by P. Sarkar and T. Iwata. Vol. 8873. LNCS. Springer, 2014, pp. 63–84. DOI: 10.1007/978-3-662-45611-8_4 (p. 135).

[BBS99] E. Biham, A. Biryukov, and A. Shamir. Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In: Advances in Cryptology – EUROCRYPT 1999. Ed. by J. Stern. Vol. 1592. LNCS. Springer, 1999, pp. 12–23. DOI: 10.1007/3-540-48910-X_2 (pp. 2, 37, 226, 252).

[BC16] C. Boura and A. Canteaut. Another View of the Division Property. In: Advances in Cryptology – CRYPTO 2016. Ed. by M. Robshaw and J. Katz. Vol. 9814. LNCS. Springer, 2016, pp. 654–682. DOI: 10.1007/978-3-662-53018-4_24 (p. 42).

[BC86] G. Brassard and C. Crépeau. Zero-Knowledge Simulation of Boolean Circuits. In: Advances in Cryptology – CRYPTO 1986. Ed. by A. M. Odlyzko. Vol. 263. LNCS. Springer, 1986, pp. 223–233. DOI: 10.1007/3-540-47721-7_16 (p. 17).

[BCBP03] A. Biryukov, C. D. Cannière, A. Braeken, and B. Preneel. A Toolbox for Cryptanalysis: Linear and Affine Equivalence Algorithms. In: Advances in Cryptology – EUROCRYPT 2003. Ed. by E. Biham. Vol. 2656. LNCS. Springer, 2003, pp. 33–50. DOI: 10.1007/3-540-39200-9_3 (p. 47).

[BCC11] C. Boura, A. Canteaut, and C. D. Cannière. Higher-Order Differential Properties of Keccak and *Luffa*. In: Fast Software Encryption – FSE 2011. Ed. by A. Joux. Vol. 6733. LNCS. Springer, 2011, pp. 252–269. DOI: 10.1007/978-3-642-21702-9_15 (pp. 42, 217, 223, 224, 271).

[BCC19] C. Boura, A. Canteaut, and D. Coggia. A General Proof Framework for Recent AES Distinguishers. In: IACR Transactions on Symmetric Cryptology 2019.1 (2019), pp. 170–191. DOI: 10.13154/tosc.v2019.i1.170-191. URL: https://tosc.iacr.org/index.php/ToSC/article/view/7401 (pp. 68, 253).

[BCG+12] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavun, M. Knezevic, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S. S. Thomsen, and T. Yalçin. PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract. In: Advances in Cryptology – ASIACRYPT 2012. Ed. by X. Wang and K. Sako. Vol. 7658. LNCS. Springer, 2012, pp. 208–225. DOI: 10.1007/978-3-642-34961-4_14 (pp. 98, 99, 242).

[BCG+13] E. Ben-Sasson, A. Chiesa, D. Genkin, E. Tromer, and M. Virza. SNARKs for C: Verifying Program Executions Succinctly and in Zero Knowledge. In: Advances in Cryptology – CRYPTO 2013. Ed. by R. Canetti and J. A. Garay. Vol. 8043. LNCS. Springer, 2013, pp. 90–108. DOI: 10.1007/978-3-642-40084-1_6 (pp. 17, 197).

[BCG+14] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized Anonymous Payments from Bitcoin. In: 2014 IEEE Symposium on Security and Privacy – SP 2014. IEEE Computer Society, 2014, pp. 459–474. DOI: 10.1109/SP.2014.36 (pp. 17, 197, 207).

[BCLR17] C. Beierle, A. Canteaut, G. Leander, and Y. Rotella. Proving Resistance Against Invariant Attacks: How to Choose the Round Constants. In: Advances in Cryptology – CRYPTO 2017. Ed. by J. Katz and H. Shacham. Vol. 10402. LNCS. Springer, 2017, pp. 647–678. DOI: 10.1007/978-3-319-63715-0_22 (pp. 49, 59, 60, 254).

[BDD+12] C. Bouillaguet, P. Derbez, O. Dunkelman, P. Fouque, N. Keller, and V. Rijmen. Low-Data Complexity Attacks on AES. In: IEEE Trans. Information Theory 58.11 (2012), pp. 7002–7017. DOI: 10.1109/TIT.2012.2207880 (pp. 45, 261).

[BDD+15] A. Bar-On, I. Dinur, O. Dunkelman, V. Lallemand, N. Keller, and B. Tsaban. Cryptanalysis of SP Networks with Partial Non-Linear Layers. In: Advances in Cryptology – EUROCRYPT 2015. Ed. by E. Oswald and M. Fischlin. Vol. 9056. LNCS. Springer, 2015, pp. 315–342. DOI: 10.1007/978-3-662-46800-5_13 (pp. 198, 240).

[BDF11] C. Bouillaguet, P. Derbez, and P. Fouque. Automatic Search of Attacks on Round-Reduced AES and Applications. In: Advances in Cryptology – CRYPTO 2011. Ed. by P. Rogaway. Vol. 6841. LNCS. Springer, 2011, pp. 169–187. DOI: 10.1007/978-3-642-22792-9_10 (p. 45).

[BDK+18] A. Bar-On, O. Dunkelman, N. Keller, E. Ronen, and A. Shamir. Improved Key Recovery Attacks on Reduced-Round AES with Practical Data and Memory Complexities. In: Advances in Cryptology – CRYPTO 2018. Ed. by H. Shacham and A. Boldyreva. Vol. 10992. LNCS. Springer, 2018, pp. 185–212. DOI: 10.1007/978-3-319-96881-0_7 (pp. 35, 110, 120).

[BDMW10] K. Browning, J. Dillon, M. McQuistan, and A. Wolfe. An APN permutation in dimension six. In: 518 (Jan. 2010), pp. 33–42 (pp. 32, 77).

[BDOZ11] R. Bendlin, I. Damgård, C. Orlandi, and S. Zakarias. Semi-homomorphic Encryption and Multiparty Computation. In: Advances in Cryptology – EUROCRYPT 2011. Ed. by K. G. Paterson. Vol. 6632. LNCS. Springer, 2011, pp. 169–188. DOI: 10.1007/978-3-642-20465-4_11 (p. 208).

[BDP00] J. Boyar, I. Damgård, and R. Peralta. Short Non-Interactive Cryptographic Proofs. In: Journal of Cryptology 13.4 (2000), pp. 449–472. DOI: 10.1007/s001450010011 (p. 17).

[BDPA] G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. Note on zero-sum distinguishers of Keccak-f. URL: http://keccak.noekeon.org/NoteZeroSum.pdf (pp. 217, 224).

[BDPA07] G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. Sponge functions. In: Ecrypt Hash Workshop 2007. 2007. URL: http://sponge.noekeon.org/SpongeFunctions.pdf (pp. 19, 200, 215).

[BDPA08] G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. On the Indifferentiability of the Sponge Construction. In: Advances in Cryptology – EUROCRYPT 2008. Ed. by N. P. Smart. Vol. 4965. LNCS. Springer, 2008, pp. 181–197. DOI: 10.1007/978-3-540-78967-3_11 (pp. 19, 20, 200, 215, 222, 224).

[BDPA11] G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. The Keccak reference. Submission to NIST's SHA-3 Competition (Round 3). 2011. URL: http://keccak.noekeon.org/Keccak-reference-3.0.pdf (p. 19).

[BEF18] D. Boneh, S. Eskandarian, and B. Fisch. Post-Quantum EPID Group Signatures from Symmetric Primitives. Cryptology ePrint Archive, Report 2018/261. https://eprint.iacr.org/2018/261. 2018 (pp. 18, 268).

[BF09] G. Bourgeois and J. Faugère. Algebraic attack on NTRU using Witt vectors and Gröbner bases. In: Journal of Mathematical Cryptology 3.3 (2009), pp. 205–214. DOI: 10.1515/JMC.2009.011 (pp. 218, 257).

[BFM88] M. Blum, P. Feldman, and S. Micali. Proving Security Against Chosen Cyphertext Attacks. In: Advances in Cryptology – CRYPTO 1988. Ed. by S. Goldwasser. Vol. 403. LNCS. Springer, 1988, pp. 256–268. DOI: 10.1007/0-387-34799-2_20 (p. 17).

[BFP12] L. Bettale, J. Faugère, and L. Perret. Solving polynomial systems over finite fields: improved analysis of the hybrid approach. In: International Symposium on Symbolic and Algebraic Computation – ISSAC 2012. Ed. by J. van der Hoeven and M. van Hoeij. ACM, 2012, pp. 67–74. DOI: 10.1145/2442829.2442843 (pp. 218, 228, 256, 257).

[BFSY05] M. Bardet, J.-c. Faugère, B. Salvy, and B.-y. Yang. Asymptotic Behaviour of the Degree of Regularity of Semi-Regular Polynomial Systems. In: The Effective Methods in Algebraic Geometry Conference (MEGA). 2005, pp. 1–14 (p. 218).

[BG11] C. Blondeau and B. Gérard. Multiple Differential Cryptanalysis: Theory and Practice. In: Fast Software Encryption – FSE 2011. Ed. by A. Joux. Vol. 6733. LNCS. Springer, 2011, pp. 35–54. DOI: 10.1007/978-3-642-21702-9_3 (p. 108).

[BGV12] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In: Innovations in Theoretical Computer Science 2012. Ed. by S. Goldwasser. ACM, 2012, pp. 309–325. DOI: 10.1145/2090236.2090262 (p. 16).

[Bih93] E. Biham. New Types of Cryptanalytic Attacks Using related Keys (Extended Abstract). In: Advances in Cryptology – EUROCRYPT 1993. Ed. by T. Helleseth. Vol. 765. LNCS. Springer, 1993, pp. 398–409. DOI: 10.1007/3-540-48285-7_34 (pp. 46, 250).

[Bih94] E. Biham. New Types of Cryptanalytic Attacks Using Related Keys. In: Journal of Cryptology 7.4 (1994), pp. 229–246. DOI: 10.1007/BF00203965 (p. 46).

[Bir04] A. Biryukov. The Boomerang Attack on 5 and 6-Round Reduced AES. In: Advanced Encryption Standard - AES 2004. Ed. by H. Dobbertin, V. Rijmen, and A. Sowa. Vol. 3373. LNCS. Springer, 2004, pp. 11–15. DOI: 10.1007/11506447_2 (pp. 34, 35, 44).

[BJV04] T. Baignères, P. Junod, and S. Vaudenay. How Far Can We Go Beyond Linear Cryptanalysis? In: Advances in Cryptology – ASIACRYPT 2004. Ed. by P. J. Lee. Vol. 3329. LNCS. Springer, 2004, pp. 432–450. DOI: 10.1007/978-3-540-30539-2_31 (p. 96).

[BK01] E. Biham and N. Keller. Cryptanalysis of Reduced Variants of Rijndael. unpublished. http://csrc.nist.gov/archive/aes/round2/conf3/papers/35-ebiham.pdf. 2001 (pp. 3, 34, 35, 37, 58, 67, 252).

[BK07] A. Biryukov and D. Khovratovich. Two New Techniques of Side-Channel Cryptanalysis. In: Cryptographic Hardware and Embedded Systems – CHES 2007. Ed. by P. Paillier and I. Verbauwhede. Vol. 4727. LNCS. Springer, 2007, pp. 195–208. DOI: 10.1007/978-3-540-74735-2_14 (p. 58).

[BK09] A. Biryukov and D. Khovratovich. Related-Key Cryptanalysis of the Full AES-192 and AES-256. In: Advances in Cryptology – ASIACRYPT 2009. Ed. by M. Matsui. Vol. 5912. LNCS. Springer, 2009, pp. 1–18. DOI: 10.1007/978-3-642-10366-7_1 (p. 46).

[BKL+07] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In: Cryptographic Hardware and Embedded Systems – CHES 2007. Ed. by P. Paillier and I. Verbauwhede. Vol. 4727. LNCS. Springer, 2007, pp. 450–466. DOI: 10.1007/978-3-540-74735-2_31 (pp. 98, 99).

[BKLT11] J. Borghoff, L. R. Knudsen, G. Leander, and S. S. Thomsen. Cryptanalysis of PRESENT-Like Ciphers with Secret S-Boxes. In: Fast Software Encryption – FSE 2011. Ed. by A. Joux. Vol. 6733. LNCS. Springer, 2011, pp. 270–289. DOI: 10.1007/978-3-642-21702-9_16 (p. 135).

[BKN09] A. Biryukov, D. Khovratovich, and I. Nikolic. Distinguisher and Related-Key Attack on the Full AES-256. In: Advances in Cryptology – CRYPTO 2009. Ed. by S. Halevi. Vol. 5677. LNCS. Springer, 2009, pp. 231–249. DOI: 10.1007/978-3-642-03356-8_14 (pp. 46, 157, 177, 178).

[BKR11] A. Bogdanov, D. Khovratovich, and C. Rechberger. Biclique Cryptanalysis of the Full AES. In: Advances in Cryptology – ASIACRYPT 2011. Ed. by D. H. Lee and X. Wang. Vol. 7073. LNCS. Springer, 2011, pp. 344–371. DOI: 10.1007/978-3-642-25385-0_19 (pp. 35, 39, 253).

[BKW93] T. Becker, H. Kredel, and V. Weispfenning. Gröbner bases: a computational approach to commutative algebra. Springer-Verlag, 1993 (pp. 217, 256).

[BLN14] C. Blondeau, G. Leander, and K. Nyberg. Differential-Linear Cryptanalysis Revisited. In: Fast Software Encryption – FSE 2014. Ed. by C. Cid and C. Rechberger. Vol. 8540. LNCS. Springer, 2014, pp. 411–430. DOI: 10.1007/978-3-662-46706-0_21 (pp. 2, 43, 50).

[BLN17] C. Blondeau, G. Leander, and K. Nyberg. Differential-Linear Cryptanalysis Revisited. In: Journal of Cryptology 30.3 (2017), pp. 859–888. DOI: 10.1007/s00145-016-9237-5. URL: https://doi.org/10.1007/s00145-016-9237-5 (pp. 2, 50).

[BLNS18] C. Boura, V. Lallemand, M. Naya-Plasencia, and V. Suder. Making the Impossible Possible. In: Journal of Cryptology 31.1 (2018), pp. 101–133. DOI: 10.1007/s00145-016-9251-7 (pp. 3, 38).

[BLNW12] A. Bogdanov, G. Leander, K. Nyberg, and M. Wang. Integral and Multidimensional Linear Distinguishers with Correlation Zero. In: Advances in Cryptology – ASIACRYPT 2012. Ed. by X. Wang and K. Sako. Vol. 7658. LNCS. Springer, 2012, pp. 244–261. DOI: 10.1007/978-3-642-34961-4_16 (p. 43).

[BN13] C. Blondeau and K. Nyberg. New Links between Differential and Linear Cryptanalysis. In: Advances in Cryptology – EUROCRYPT 2013. Ed. by T. Johansson and P. Q. Nguyen. Vol. 7881. LNCS. Springer, 2013, pp. 388–404. DOI: 10.1007/978-3-642-38348-9_24 (p. 43).

[BN14] C. Blondeau and K. Nyberg. Links between Truncated Differential and Multidimensional Linear Properties of Block Ciphers and Underlying Attack Complexities. In: Advances in Cryptology – EUROCRYPT 2014. Ed. by P. Q. Nguyen and E. Oswald. Vol. 8441. LNCS. Springer, 2014, pp. 165–182. DOI: 10.1007/978-3-642-55220-5_10 (p. 43).

[BNS14] C. Boura, M. Naya-Plasencia, and V. Suder. Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock and Simon. In: Advances in Cryptology – ASIACRYPT 2014. Ed. by P. Sarkar and T. Iwata. Vol. 8873. LNCS. Springer, 2014, pp. 179–199. DOI: 10.1007/978-3-662-45611-8_10 (p. 226).

[BPP00] J. Boyar, R. Peralta, and D. Pochuev. On the multiplicative complexity of Boolean functions over the basis (cap, +, 1). In: Theor. Comput. Sci. 235.1 (2000), pp. 43–57. DOI: 10.1016/S0304-3975(99)00182-6 (p. 197).

[BPW15] C. Blondeau, T. Peyrin, and L. Wang. Known-Key Distinguisher on Full PRESENT. In: Advances in Cryptology – CRYPTO 2015. Ed. by R. Gennaro and M. Robshaw. Vol. 9215. LNCS. Springer, 2015, pp. 455–474. DOI: 10.1007/978-3-662-47989-6_22 (p. 157).

[BS01] A. Biryukov and A. Shamir. Structural Cryptanalysis of SASAS. In: Advances in Cryptology – EUROCRYPT 2001. Ed. by B. Pfitzmann. Vol. 2045. LNCS. Springer, 2001, pp. 394–405. DOI: 10.1007/3-540-44987-6_24 (pp. 4, 135, 136).

[BS10] A. Biryukov and A. Shamir. Structural Cryptanalysis of SASAS. In: Journal of Cryptology 23.4 (2010), pp. 505–518. DOI: 10.1007/s00145-010-9062-1 (pp. 4, 135, 136).

[BS90] E. Biham and A. Shamir. Differential Cryptanalysis of DES-like Cryptosystems. In: Advances in Cryptology – CRYPTO 1990. Ed. by A. Menezes and S. A. Vanstone. Vol. 537. LNCS. Springer, 1990, pp. 2–21. DOI: 10.1007/3-540-38424-3_1 (pp. 2, 27, 108, 203, 225, 250).

[BS91] E. Biham and A. Shamir. Differential Cryptanalysis of DES-like Cryptosystems. In: Journal of Cryptology 4.1 (1991), pp. 3–72. DOI: 10.1007/BF00630563 (pp. 2, 27).

[BS92] E. Biham and A. Shamir. Differential Cryptanalysis of the Full 16-Round DES. In: Advances in Cryptology – CRYPTO 1992. Ed. by E. F. Brickell. Vol. 740. LNCS. Springer, 1992, pp. 487–496. DOI: 10.1007/3-540-48071-4_34 (p. 27).

[BS93] E. Biham and A. Shamir. Differential Cryptanalysis of the Data Encryption Standard. Springer, 1993. DOI: 10.1007/978-1-4613-9314-6 (pp. 2, 27, 203, 225, 250).

[BV05] T. Baignères and S. Vaudenay. Proving the Security of AES Substitution-Permutation Network. In: Selected Areas in Cryptography - SAC 2005. Ed. by B. Preneel and S. E. Tavares. Vol. 3897. LNCS. Springer, 2005, pp. 65–81. DOI: 10.1007/11693383_5 (p. 135).

[BWP05] A. Braeken, C. Wolf, and B. Preneel. Normality of Vectorial Functions. In: Cryptography and Coding – 10th IMA International Conference 2005. Ed. by N. P. Smart. Vol. 3796. LNCS. Springer, 2005, pp. 186–200. DOI: 10.1007/11586821_13 (pp. 62, 191).

[Car10] Carlet, Claude. Boolean Functions for Cryptography and Error-Correcting Codes. In: Boolean Models and Methods in Mathematics, Computer Science, and Engineering. Ed. by Crama, Yves and Hammer, Peter L.Editors. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2010, pp. 257–397. DOI: 10.1017/CBO9780511780448.011 (p. 31).

[CCF+16] A. Canteaut, S. Carpov, C. Fontaine, T. Lepoint, M. Naya-Plasencia, P. Paillier, and R. Sirdey. Stream Ciphers: A Practical Solution for Efficient Homomorphic-Ciphertext Compression. In: Fast Software Encryption – FSE 2016. Ed. by T. Peyrin. Vol. 9783. LNCS. Springer, 2016, pp. 313–333. DOI: 10.1007/978-3-662-52993-5_16 (p. 198).

[CCZ98] C. Carlet, P. Charpin, and V. A. Zinoviev. Codes, Bent Functions and Permutations Suitable For DES-like Cryptosystems. In: Des. Codes Cryptography 15.2 (1998), pp. 125–156. DOI: 10.1023/A:1008344232130 (pp. 31, 41, 251).

[CDG+17] M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, and G. Zaverucha. Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security – CCS 2017. Ed. by B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu. ACM, 2017, pp. 1825–1842. DOI: 10.1145/3133956.3133997 (pp. 17, 237, 263, 268).

[CDK+18] B. Cogliati, Y. Dodis, J. Katz, J. Lee, J. P. Steinberger, A. Thiruvengadam, and Z. Zhang. Provable Security of (Tweakable) Block Ciphers Based on Substitution-Permutation Networks. In: CRYPTO 2018. Ed. by H. Shacham and A. Boldyreva. Vol. 10991. LNCS. Springer, 2018, pp. 722–753. DOI: 10.1007/978-3-319-96884-1_24 (p. 271).

[CFG+17] C. Chaigneau, T. Fuhr, H. Gilbert, J. Jean, and J.-R. Reinhard. Cryptanalysis of NORX v2.0. In: IACR Transactions on Symmetric Cryptology 2017.1 (Mar. 2017), pp. 156–174. DOI: 10.13154/tosc.v2017.i1.156-174. URL: https://tosc.iacr.org/index.php/ToSC/article/view/589 (p. 59).

[CFH+15] C. Costello, C. Fournet, J. Howell, M. Kohlweiss, B. Kreuter, M. Naehrig, B. Parno, and S. Zahur. Geppetto: Versatile Verifiable Computation. In: 2015 IEEE Symposium on Security and Privacy – SP 2015. IEEE Computer Society, 2015, pp. 253–270. DOI: 10.1109/SP.2015.23 (p. 207).

[CGH04] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In: Journal ACM 51.4 (2004), pp. 557–594. DOI: 10.1145/1008731.1008734 (pp. 158, 176).

[CKK+02] J. H. Cheon, M. Kim, K. Kim, L. Jung-Yeun, and S. Kang. Improved Impossible Differential Cryptanalysis of Rijndael and Crypton. In: Information Security and Cryptology – ICISC 2001. Ed. by K. Kim. Vol. 2288. LNCS. Springer, 2002, pp. 39–49. DOI: 10.1007/3-540-45861-1_4 (pp. 35, 37).

[CL05] C. Cid and G. Leurent. An Analysis of the XSL Algorithm. In: Advances in Cryptology – ASIACRYPT 2005. Ed. by B. K. Roy. Vol. 3788. LNCS. Springer, 2005, pp. 333–352. DOI: 10.1007/11593447_18 (p. 40).

[CLT14] J. Coron, T. Lepoint, and M. Tibouchi. Scale-Invariant Fully Homomorphic Encryption over the Integers. In: Public-Key Cryptography – PKC 2014. Ed. by H. Krawczyk. Vol. 8383. LNCS. Springer, 2014, pp. 311–328. DOI: 10.1007/978-3-642-54631-0_18 (p. 16).

[CMR05] C. Cid, S. Murphy, and M. J. B. Robshaw. Small Scale Variants of the AES. In: Fast Software Encryption - FSE 2005. Ed. by H. Gilbert and H. Handschuh. Vol. 3557. LNCS. Springer, 2005, pp. 145–162. DOI: 10.1007/11502760_10 (pp. 90, 99, 113, 119, 130, 153, 155).

[CP02] N. Courtois and J. Pieprzyk. Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. In: Advances in Cryptology – ASIACRYPT 2002. Ed. by Y. Zheng. Vol. 2501. LNCS. Springer, 2002, pp. 267–287. DOI: 10.1007/3-540-36178-2_17 (p. 39).

[CSCW17] T. Cui, L. Sun, H. Chen, and M. Wang. Statistical Integral Distinguisher with Multi-structure and Its Application on AES. In: Information Security and Privacy – ACISP 2017. Ed. by J. Pieprzyk and S. Suriadi. Vol. 10342. LNCS. Springer, 2017, pp. 402–420. DOI: 10.1007/978-3-319-60055-0_21 (pp. 164, 170, 171).

[CV94] F. Chabaud and S. Vaudenay. Links Between Differential and Linear Cryptanalysis. In: Advances in Cryptology – EUROCRYPT 1994. Ed. by A. D. Santis. Vol. 950. LNCS. Springer, 1994, pp. 356–365. DOI: 10.1007/BFb0053450 (pp. 31, 43).

[Dae95] J. Daemen. Cipher and Hash Function Design. Strategies based on linear and differential cryptanalysis. PhD Thesis. Katholieke Universiteit Leuven. 1995. URL: https://www.esat.kuleuven.be/cosic/publications/thesis-6.pdf (p. 32).

[DDKS12] I. Dinur, O. Dunkelman, N. Keller, and A. Shamir. Efficient Dissection of Composite Problems, with Applications to Cryptanalysis, Knapsacks, and Combinatorial Search Problems. In: Advances in Cryptology – CRYPTO 2012. Ed. by R. Safavi-Naini and R. Canetti. Vol. 7417. LNCS. Springer, 2012, pp. 719–740. DOI: 10.1007/978-3-642-32009-5_42 (p. 120).

[DEG+18] C. Dobraunig, M. Eichlseder, L. Grassi, V. Lallemand, G. Leander, E. List, F. Mendel, and C. Rechberger. Rasta: A Cipher with Low ANDdepth and Few ANDs per Bit. Ed. by H. Shacham and A. Boldyreva. 2018. DOI: 10.1007/978-3-319-96884-1_22 (pp. 198, 242).

[DEM15] C. Dobraunig, M. Eichlseder, and F. Mendel. Higher-Order Cryptanalysis of LowMC. In: ICISC 2015. Ed. by S. Kwon and A. Yun. Vol. 9558. LNCS. Springer, 2015, pp. 87–101. DOI: 10.1007/978-3-319-30840-1_6 (p. 241).

[Der13] P. Derbez. Meet-in-the-Middle Attacks on AES. (Attaques par Rencontre par le Milieu sur l'AES). PhD Thesis. École Normale Supérieure, Paris, France. 2013. URL: https://tel.archives-ouvertes.fr/tel-00918146 (p. 35).

[DF13] P. Derbez and P. Fouque. Exhausting Demirci-Selçuk Meet-in-the-Middle Attacks Against Reduced-Round AES. In: Fast Software Encryption – FSE 2013. Ed. by S. Moriai. Vol. 8424. LNCS. Springer, 2013, pp. 541–560. DOI: 10.1007/978-3-662-43933-3_28 (pp. 35, 39, 252).

[DFJ12] P. Derbez, P. Fouque, and J. Jean. Faster Chosen-Key Distinguishers on Reduced-Round AES. In: Progress in Cryptology – INDOCRYPT 2012. Ed. by S. D. Galbraith and M. Nandi. Vol. 7668. LNCS. Springer, 2012, pp. 225–243. DOI: 10.1007/978-3-642-34931-7_14 (pp. 177, 178).

[DFJ13] P. Derbez, P. Fouque, and J. Jean. Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting. In: Advances in Cryptology – EUROCRYPT 2013. Ed. by T. Johansson and P. Q. Nguyen. Vol. 7881. LNCS. Springer, 2013, pp. 371–387. DOI: 10.1007/978-3-642-38348-9_23 (p. 39).

[DGKN09] I. Damgård, M. Geisler, M. Krøigaard, and J. B. Nielsen. Asynchronous Multiparty Computation: Theory and Implementation. In: Public Key Cryptography – PKC 2009. Ed. by S. Jarecki and G. Tsudik. Vol. 5443. LNCS. Springer, 2009, pp. 160–179. DOI: 10.1007/978-3-642-00468-1_10 (p. 208).

[DK10] I. Damgård and M. Keller. Secure Multiparty AES. In: Financial Cryptography and Data Security – FC 2010. Ed. by R. Sion. Vol. 6052. LNCS. Springer, 2010, pp. 367–374. DOI: 10.1007/978-3-642-14577-3_31 (p. 16).

[DKL+12] I. Damgård, M. Keller, E. Larraia, C. Miles, and N. P. Smart. Implementing AES via an Actively/Covertly Secure Dishonest-Majority MPC Protocol. In: Security and Cryptography for Networks – SCN 2012. Ed. by I. Visconti and R. D. Prisco. Vol. 7485. LNCS. Springer, 2012, pp. 241–263. DOI: 10.1007/978-3-642-32928-9_14 (p. 16).

[DKP+19] I. Dinur, D. Kales, A. Promitzer, S. Ramacher, and C. Rechberger. Linear Equivalence of Block Ciphers with Partial Non-Linear Layers: Application to LowMC. In: EUROCRYPT. to Appear. 2019 (pp. 246, 269).

[DKR97] J. Daemen, L. R. Knudsen, and V. Rijmen. The Block Cipher Square. In: Fast Software Encryption – FSE 1997. Ed. by E. Biham. Vol. 1267. LNCS. Springer, 1997, pp. 149–165. DOI: 10.1007/BFb0052343 (pp. 2, 34, 67, 120, 169, 254, 271).

[DKS10] O. Dunkelman, N. Keller, and A. Shamir. Improved Single-Key Attacks on 8-Round AES-192 and AES-256. In: Advances in Cryptology – ASIACRYPT 2010. Ed. by M. Abe. Vol. 6477. LNCS. Springer, 2010, pp. 158–176. DOI: 10.1007/978-3-642-17373-8_10 (p. 38).

[DLMW15] I. Dinur, Y. Liu, W. Meier, and Q. Wang. Optimized Interpolation Attacks on LowMC. In: Advances in Cryptology – ASIACRYPT 2015. Ed. by T. Iwata and J. H. Cheon. Vol. 9453. LNCS. Springer, 2015, pp. 535–560. DOI: 10.1007/978-3-662-48800-3_22 (p. 241).

[DM95] D. W. Davies and S. Murphy. Pairs and Triplets of DES S-Boxes. In: Journal of Cryptology 8.1 (1995), pp. 1–25. DOI: 10.1007/BF00204799 (p. 47).

[DPAR00] J. Daemen, M. Peeters, G. V. Assche, and V. Rijmen. Nessie Proposal: the block cipher NOEKEON. Nessie submission. http://gro.noekeon.org/. 2000 (pp. 98, 99).

[DPSZ12] I. Damgård, V. Pastro, N. P. Smart, and S. Zakarias. Multiparty Computation from Somewhat Homomorphic Encryption. In: Advances in Cryptology – CRYPTO 2012. Ed. by R. Safavi-Naini and R. Canetti. Vol. 7417. LNCS. Springer, 2012, pp. 643–662. DOI: 10.1007/978-3-642-32009-5_38 (p. 208).

[DR00] J. Daemen and V. Rijmen. Rijndael for AES. In: AES Candidate Conference. 2000, pp. 343–348 (p. 25).

[DR01] J. Daemen and V. Rijmen. The Wide Trail Design Strategy. In: Cryptography and Coding – IMA International Conference. Ed. by B. Honary. Vol. 2260. LNCS. Springer, 2001, pp. 222–238. DOI: 10.1007/3-540-45325-3_20 (pp. 31, 239).

[DR02a] J. Daemen and V. Rijmen. Security of a Wide Trail Design. In: Progress in Cryptology – INDOCRYPT 2002. Ed. by A. Menezes and P. Sarkar. Vol. 2551. LNCS. Springer, 2002, pp. 1–11. DOI: 10.1007/3-540-36231-2_1 (pp. 31, 239).

[DR02b] J. Daemen and V. Rijmen. The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography. Springer, 2002. DOI: 10.1007/978-3-662-04722-4 (pp. 1, 2, 25, 34, 35, 239).

[DR06] J. Daemen and V. Rijmen. Understanding Two-Round Differentials in AES. In: Security and Cryptography for Networks – SCN 2006. Ed. by R. D. Prisco and M. Yung. Vol. 4116. LNCS. Springer, 2006, pp. 78–94. DOI: 10.1007/11832072_6 (pp. 33, 34, 47).

[DR98] J. Daemen and V. Rijmen. The Block Cipher Rijndael. In: Smart Card Research and Applications – CARDIS 1998. Ed. by J. Quisquater and B. Schneier. Vol. 1820. LNCS. Springer, 1998, pp. 277–284. DOI: 10.1007/10721064_26 (pp. 25, 34).

[DRS18a] D. Derler, S. Ramacher, and D. Slamanig. Generic Double-Authentication Preventing Signatures and a Post-quantum Instantiation. In: Provable Security – ProvSec 2018. Ed. by J. Baek, W. Susilo, and J. Kim. Vol. 11192. LNCS. Springer, 2018, pp. 258–276. DOI: 10.1007/978-3-030-01446-9_15 (pp. 18, 268).

[DRS18b] D. Derler, S. Ramacher, and D. Slamanig. Post-Quantum Zero-Knowledge Proofs for Accumulators with Applications to Ring Signatures from Symmetric-Key Primitives. In: Post-Quantum Cryptography - PQCrypto 2018. Ed. by T. Lange and R. Steinwandt. Vol. 10786. LNCS. Springer, 2018, pp. 419–440. DOI: 10.1007/978-3-319-79063-3_20 (pp. 18, 268).

[DS08] H. Demirci and A. A. Selçuk. A Meet-in-the-Middle Attack on 8-Round AES. In: Fast Software Encryption – FSE 2008. Ed. by K. Nyberg. Vol. 5086. LNCS. Springer, 2008, pp. 116–126. DOI: 10.1007/978-3-540-71039-4_7 (p. 38).

[DTÇB09] H. Demirci, İ. Taşkın, M. Çoban, and A. Baysal. Improved Meet-in-the-Middle Attacks on AES. In: Progress in Cryptology – INDOCRYPT 2009. Ed. by B. Roy and N. Sendrier. Vol. 5922. LNCS. Springer, 2009, pp. 144–156. DOI: 10.1007/978-3-642-10628-6_10 (p. 38).

[Eic18] M. Eichlseder. Differential Cryptanalysis of Symmetric Primitives. PhD Thesis. IAIK, Graz University of Technology (Austria). 2018 (p. 9).

[FJP13] P. Fouque, J. Jean, and T. Peyrin. Structural Evaluation of AES and Chosen-Key Distinguisher of 9-Round AES-128. In: Advances in Cryptology – CRYPTO 2013. Ed. by R. Canetti and J. A. Garay. Vol. 8042. LNCS. Springer, 2013, pp. 183–203. DOI: 10.1007/978-3-642-40041-4_11 (pp. 177, 178).

[FKL+00] N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. A. Wagner, and D. Whiting. Improved Cryptanalysis of Rijndael. In: Fast Software Encryption – FSE 2000. Ed. by B. Schneier. Vol. 1978. LNCS. Springer, 2000, pp. 213–230. DOI: 10.1007/3-540-44706-7_15 (pp. 35, 36).

[FS86] A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Advances in Cryptology – CRYPTO 1986. Ed. by A. M. Odlyzko. Vol. 263. LNCS. Springer, 1986, pp. 186–194. DOI: 10.1007/3-540-47721-7_12 (p. 18).

[FSW01] N. Ferguson, R. Schroeppel, and D. Whiting. A Simple Algebraic Representation of Rijndael. In: Selected Areas in Cryptography – SAC 2001. Ed. by S. Vaudenay and A. M. Youssef. Vol. 2259. LNCS. Springer, 2001, pp. 103–111. DOI: 10.1007/3-540-45537-X_8 (pp. 39, 47).

[GC94] H. Gilbert and P. Chauvaud. A Chosen Plaintext Attack of the 16-round Khufu Cryptosystem. In: Advances in Cryptology – CRYPTO 1994. Ed. by Y. Desmedt. Vol. 839. LNCS. Springer, 1994, pp. 359–368. DOI: 10.1007/3-540-48658-5_33 (p. 135).

[GGNS13] B. Gérard, V. Grosso, M. Naya-Plasencia, and F. Standaert. Block Ciphers That Are Easier to Mask: How Far Can We Go? In: Cryptographic Hardware and Embedded Systems – CHES 2013. Ed. by G. Bertoni and J. Coron. Vol. 8086. LNCS. Springer, 2013, pp. 383–399. DOI: 10.1007/978-3-642-40349-1_22 (pp. 3, 198, 240).

[Gil14] H. Gilbert. A Simplified Representation of AES. In: Advances in Cryptology – ASIACRYPT 2014. Ed. by P. Sarkar and T. Iwata. Vol. 8873. LNCS. Springer, 2014, pp. 200–222. DOI: 10.1007/978-3-662-45611-8_11 (pp. 4, 48, 159, 164–166, 168–172, 174, 175).

[GJN+16] J. Guo, J. Jean, I. Nikolic, K. Qiao, Y. Sasaki, and S. Sim. Invariant Subspace Attack Against Midori64 and The Resistance Criteria for S-box Designs. In: IACR Transactions on Symmetric Cryptology 2016.1 (Dec. 2016), pp. 33–56. DOI: 10.13154/tosc.v2016.i1.33-56. URL: https://tosc.iacr.org/index.php/ToSC/article/view/534 (p. 49).

[GLR+18] L. Grassi, G. Leander, C. Rechberger, C. Tezcan, and F. Wiemer. Weak-Key Subspace Trails and Applications to AES. In Submission. 2018 (pp. 52, 59, 61, 177–179, 182, 183, 186).

[GLR+19] L. Grassi, R. Lueftenegger, S. Ramacher, C. Rechberger, D. Rotaru, and M. Schofnegger. On a Generalization of Substitution-Permutation Networks: The HADES Design Strategy. In Submission. 2019 (pp. 6, 239, 241, 243, 245, 249, 255, 259, 266–268).

[GM16] S. Gueron and N. Mouha. Simpira v2: A Family of Efficient Permutations Using the AES Round Function. In: Advances in Cryptology – ASIACRYPT 2016. Ed. by J. H. Cheon and T. Takagi. Vol. 10031. LNCS. 2016, pp. 95–125. DOI: 10.1007/978-3-662-53887-6_4 (p. 3).

[GMO16] I. Giacomelli, J. Madsen, and C. Orlandi. ZKBoo: Faster Zero-Knowledge for Boolean Circuits. In: 25th USENIX Security Symposium – USENIX Security 2016. Ed. by T. Holz and S. Savage. USENIX Association, 2016, pp. 1069–1083 (pp. 17, 18, 268).

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract). In: ACM Symposium on Theory of Computing – STOC 1985. Ed. by R. Sedgewick. ACM, 1985, pp. 291–304. DOI: 10.1145/22145.22178 (p. 17).

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. In: SIAM J. Comput. 18.1 (1989), pp. 186–208. DOI: 10.1137/0218012 (p. 17).

[GNL11] Z. Gong, S. Nikova, and Y. W. Law. KLEIN: A New Family of Lightweight Block Ciphers. In: RFID. Security and Privacy - RFIDSec 2011. Ed. by A. Juels and C. Paar. Vol. 7055. LNCS. Springer, 2011, pp. 1–18. DOI: 10.1007/978-3-642-25286-0_1 (pp. 98, 99).

[GNPW13] J. Guo, I. Nikolic, T. Peyrin, and L. Wang. Cryptanalysis of Zorro. Cryptology ePrint Archive, Report 2013/713. https://eprint.iacr.org/2013/713. 2013 (pp. 59, 240).

[GP10] H. Gilbert and T. Peyrin. Super-Sbox Cryptanalysis: Improved Attacks for AES-Like Permutations. In: Fast Software Encryption – FSE 2010. Ed. by S. Hong and T. Iwata. Vol. 6147. LNCS. Springer, 2010, pp. 365–383. DOI: 10.1007/978-3-642-13858-4_21 (pp. 47, 164, 165, 170, 172, 177).

[GPPR11] J. Guo, T. Peyrin, A. Poschmann, and M. J. B. Robshaw. The LED Block Cipher. In: Cryptographic Hardware and Embedded Systems – CHES 2011. Ed. by B. Preneel and T. Takagi. Vol. 6917. LNCS. Springer, 2011, pp. 326–341. DOI: 10.1007/978-3-642-23951-9_22 (p. 3).

[GR17] L. Grassi and C. Rechberger. New and Old Limits for AES Known-Key Distinguishers. Cryptology ePrint Archive, Report 2017/255. In Submission. 2017. URL: https://eprint.iacr.org/2017/255 (pp. 164, 170–176).

[GR18] L. Grassi and C. Rechberger. New Rigorous Analysis of Truncated Differentials for 5-round AES. Cryptology ePrint Archive, Report 2018/182. https://eprint.iacr.org/2018/182. 2018 (pp. 34, 35, 67, 74, 85, 86, 90, 98, 99).

[Gra17a] L. Grassi. MixColumns Properties and Attacks on (round-reduced) AES with a Single Secret S-Box. Cryptology ePrint Archive, Report 2017/1200. https://eprint.iacr.org/2017/1200. 2017 (pp. 152, 155).

[Gra17b] L. Grassi. Mixture Differential Cryptanalysis and Structural Truncated Differential Attacks on round-reduced AES. Cryptology ePrint Archive, Report 2017/832. https://eprint.iacr.org/2017/832. 2017 (pp. 34, 35, 75, 107, 109–111, 123, 127–129).

[Gra18a] L. Grassi. MixColumns Properties and Attacks on (Round-Reduced) AES with a Single Secret S-Box. In: Topics in Cryptology - CT-RSA 2018. Ed. by N. P. Smart. Vol. 10808. LNCS. Springer, 2018, pp. 243–263. DOI: 10.1007/978-3-319-76953-0_13 (pp. 136, 137, 140, 141, 148, 149, 151, 154).

[Gra18b] L. Grassi. Mixture Differential Cryptanalysis: a New Approach to Distinguishers and Attacks on round-reduced AES. In: IACR Transaction on Symmetric Cryptology 2018.2 (2018), pp. 133–160. DOI: 10.13154/tosc.v2018.i2.133-160. URL: https://doi.org/10.13154/tosc.v2018.i2.133-160 (pp. 34, 35, 75, 107, 110, 111, 253).

[GRR+16] L. Grassi, C. Rechberger, D. Rotaru, P. Scholl, and N. P. Smart. MPC-Friendly Symmetric Key Primitives. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Ed. by E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi. ACM, 2016, pp. 430–443. DOI: 10.1145/2976749.2978332 (pp. 207–209, 267).

[GRR16] L. Grassi, C. Rechberger, and S. Rønjom. Subspace Trail Cryptanalysis and its Applications to AES. In: IACR Trans. Symmetric Cryptol. 2016.2 (2016), pp. 192–225. DOI: 10.13154/tosc.v2016.i2.192-225 (pp. 45–47, 49, 50, 53–55, 57, 136–139, 144, 148, 149, 172).

[GRR17] L. Grassi, C. Rechberger, and S. Rønjom. A New Structural-Differential Property of 5-Round AES. In: Advances in Cryptology – EUROCRYPT 2017. Ed. by J. Coron and J. B. Nielsen. Vol. 10211. LNCS. Springer, 2017, pp. 289–317. DOI: 10.1007/978-3-319-56614-6_10 (pp. 34, 35, 67–69, 73, 75, 86, 107, 108, 110, 113–115, 120, 177, 179, 253).

[HCGW18] K. Hu, T. Cui, C. Gao, and M. Wang. Towards Key-Dependent Integral and Impossible Differential Distinguishers on 5-Round AES. In: Selected Areas in Cryptography – SAC 2018. Ed. by C. Cid and M. J. J. Jr. Vol. 11349. LNCS. Springer, 2018, pp. 139–162 (p. 136).

[HS14] S. Halevi and V. Shoup. Algorithms in HElib. In: Advances in Cryptology – CRYPTO 2014. Ed. by J. A. Garay and R. Gennaro. Vol. 8616. LNCS. Springer, 2014, pp. 554–571. DOI: 10.1007/978-3-662-44371-2_31 (p. 16).

[IBM] IBM. The MARS Encryption Algorithm. Submitted to AES Process (p. 242).

[IKOS09] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Zero-Knowledge Proofs from Secure Multiparty Computation. In: SIAM J. Comput. 39.3 (2009), pp. 1121–1152. DOI: 10.1137/080725398 (p. 17).

[Jea16a] J. Jean. Cryptanalysis of Haraka. In: IACR Transactions on Symmetric Cryptology 2016.1 (Dec. 2016), pp. 1–12. DOI: 10.13154/tosc.v2016.i1.1-12. URL: https://tosc.iacr.org/index.php/ToSC/article/view/531 (p. 49).

[Jea16b] J. Jean. TikZ for Cryptographers. https://www.iacr.org/authors/tikz/. 2016 (pp. 11, 12, 26).

[JK97] T. Jakobsen and L. R. Knudsen. The Interpolation Attack on Block Ciphers. In: Fast Software Encryption – FSE 1997. Ed. by E. Biham. Vol. 1267. LNCS. Springer, 1997, pp. 28–40. DOI: 10.1007/BFb0052332 (pp. 2, 5, 39, 197, 201, 202, 221, 240, 249, 254).

[JKO13] M. Jawurek, F. Kerschbaum, and C. Orlandi. Zero-knowledge using garbled circuits: how to prove non-algebraic statements efficiently. In: 2013 ACM SIGSAC Conference on Computer and Communications Security – CCS 2013. Ed. by A. Sadeghi, V. D. Gligor, and M. Yung. ACM, 2013, pp. 955–966. DOI: 10.1145/2508859.2516662 (p. 17).

[JNP13] J. Jean, M. Naya-Plasencia, and T. Peyrin. Multiple Limited-Birthday Distinguishers and Applications. In: Selected Areas in Cryptography – SAC 2013. Ed. by T. Lange, K. E. Lauter, and P. Lisonek. Vol. 8282. LNCS. Springer, 2013, pp. 533–550. DOI: 10.1007/978-3-662-43414-7_27 (pp. 164, 165, 170, 171, 177).

[Ker83] A. Kerckhoffs. La cryptographie militaire. In: Journal des sciences militaires IX (1883), pp. 5–83 (p. 9).

[KKW18] J. Katz, V. Kolesnikov, and X. Wang. Improved Non-Interactive Zero Knowledge with Applications to Post-Quantum Signatures. Cryptology ePrint Archive, Report 2018/475. https://eprint.iacr.org/2018/475. 2018 (pp. 17, 18, 268).

[KLMR16] S. Kölbl, M. M. Lauridsen, F. Mendel, and C. Rechberger. Haraka v2 - Efficient Short-Input Hashing for Post-Quantum Applications. In: IACR Trans. Symmetric Cryptol. 2016.2 (2016), pp. 1–29. URL: https://doi.org/10.13154/tosc.v2016.i2.1-29 (p. 3).

[KLPS17] K. Khoo, E. Lee, T. Peyrin, and S. M. Sim. Human-readable Proof of the Related-Key Security of AES-128. In: IACR Transactions of Symmetric Cryptology 2017.2 (2017), pp. 59–83. DOI: 10.13154/tosc.v2017.i2.59-83. URL: https://doi.org/10.13154/tosc.v2017.i2.59-83 (pp. 59, 60).

[KMT01] L. Keliher, H. Meijer, and S. Tavares. Improving the Upper Bound on the Maximum Average Linear Hull Probability for Rijndael. In: Selected Areas in Cryptography – SAC 2001. Ed. by S. Vaudenay and A. M. Youssef. Vol. 2259. LNCS. Springer, 2001, pp. 112–128. DOI: 10.1007/3-540-45537-X_9 (p. 33).

[Knu94] L. R. Knudsen. Truncated and Higher Order Differentials. In: Fast Software Encryption – FSE 1994. Ed. by B. Preneel. Vol. 1008. LNCS. Springer, 1994, pp. 196–211. DOI: 10.1007/3-540-60590-8_16 (pp. 2, 36, 40, 108, 124, 202, 222, 225, 240, 251, 260).

[Knu98] L. R. Knudsen. DEAL - A 128-bit Block Cipher. Technical Report 151, Department of Informatics, University of Bergen, Norway. Feb. 1998 (pp. 2, 37, 108, 226, 252).

[KPP+17] D. Kales, L. Perrin, A. Promitzer, S. Ramacher, and C. Rechberger. Improvements to the Linear Operations of LowMC: A Faster Picnic. Cryptology ePrint Archive, Report 2017/1148. https://eprint.iacr.org/2017/1148. 2017 (p. 246).

[KR07] L. R. Knudsen and V. Rijmen. Known-Key Distinguishers for Some Block Ciphers. In: Advances in Cryptology – ASIACRYPT 2007. Ed. by K. Kurosawa. Vol. 4833. LNCS. Springer, 2007, pp. 315–324. DOI: 10.1007/978-3-540-76900-2_19 (pp. 157, 164, 250).

[KR11] L. R. Knudsen and M. Robshaw. The Block Cipher Companion. Information Security and Cryptography. Springer, 2011. DOI: 10.1007/978-3-642-17342-4 (pp. 5, 9, 198, 238).

[KRS12] D. Khovratovich, C. Rechberger, and A. Savelieva. Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 Family. In: Fast Software Encryption – FSE 2012. Ed. by A. Canteaut. Vol. 7549. LNCS. Springer, 2012, pp. 244–263. DOI: 10.1007/978-3-642-34047-5_15 (p. 39).

[KS08] V. Kolesnikov and T. Schneider. Improved Garbled Circuit: Free XOR Gates and Applications. In: Automata, Languages and Programming – ICALP 2008. Ed. by L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz. Vol. 5126. LNCS. Springer, 2008, pp. 486–498. DOI: 10.1007/978-3-540-70583-3_40 (p. 17).

[KSS13] M. Keller, P. Scholl, and N. P. Smart. An architecture for practical actively secure MPC with dishonest majority. In: 2013 ACM SIGSAC Conference on Computer and Communications Security – CCS 2013. Ed. by A. Sadeghi, V. D. Gligor, and M. Yung. ACM, 2013, pp. 549–560. DOI: 10.1145/2508859.2516744 (pp. 235, 267).

[KW02] L. R. Knudsen and D. A. Wagner. Integral Cryptanalysis. In: Fast Software Encryption – FSE 2002. Ed. by J. Daemen and V. Rijmen. Vol. 2365. LNCS. Springer, 2002, pp. 112–127. DOI: 10.1007/3-540-45661-9_9 (pp. 2, 34, 67, 120, 169).

[LAAZ11] G. Leander, M. A. Abdelraheem, H. AlKhzaimi, and E. Zenner. A Cryptanalysis of PRINTcipher: The Invariant Subspace Attack. In: Advances in Cryptology – CRYPTO 2011. Ed. by P. Rogaway. Vol. 6841. LNCS. Springer, 2011, pp. 206–221. DOI: 10.1007/978-3-642-22792-9_12 (pp. 3, 48, 49, 254).

[Lai94] X. Lai. Higher Order Derivatives and Differential Cryptanalysis. In: Communications and Cryptography: Two Sides of One Tapestry. Ed. by R. E. Blahut, D. J. Costello, U. Maurer, and T. Mittelholzer. Springer US, 1994, pp. 227–233. DOI: 10.1007/978-1-4615-2694-0_23 (pp. 40, 41).

[LDKK08] J. Lu, O. Dunkelman, N. Keller, and J. Kim. New Impossible Differential Attacks on AES. In: Progress in Cryptology – INDOCRYPT 2008. Ed. by D. R. Chowdhury, V. Rijmen, and A. Das. Vol. 5365. LNCS. Springer, 2008, pp. 279–293. DOI: 10.1007/978-3-540-89754-5_22 (p. 38).

[Lea11] G. Leander. On Linear Hulls, Statistical Saturation Attacks, PRESENT and a Cryptanalysis of PUFFIN. In: Advances in Cryptology – EUROCRYPT 2011. Ed. by K. G. Paterson. Vol. 6632. LNCS. Springer, 2011, pp. 303–322. DOI: 10.1007/978-3-642-20465-4_18 (p. 43).

[LH94] S. K. Langford and M. E. Hellman. Differential-Linear Cryptanalysis. In: Advances in Cryptology – CRYPTO 1994. Ed. by Y. Desmedt. Vol. 839. LNCS. Springer, 1994, pp. 17–25. DOI: 10.1007/3-540-48658-5_3 (p. 2).

[LMM91] X. Lai, J. L. Massey, and S. Murphy. Markov Ciphers and Differential Cryptanalysis. In: Advances in Cryptology – EUROCRYPT 1991. Ed. by D. W. Davies. Vol. 547. LNCS. Springer, 1991, pp. 17–38. DOI: 10.1007/3-540-46416-6_2 (p. 29).

[LMR+09] M. Lamberger, F. Mendel, C. Rechberger, V. Rijmen, and M. Schläffer. Rebound Distinguishers: Results on the Full Whirlpool Compression Function. In: Advances in Cryptology – ASIACRYPT 2009. Ed. by M. Matsui. Vol. 5912. LNCS. Springer, 2009, pp. 126–143. DOI: 10.1007/978-3-642-10366-7_8 (pp. 47, 164).

[LMR15] G. Leander, B. Minaud, and S. Rønjom. A Generic Approach to Invariant Subspace Attacks: Cryptanalysis of Robin, iSCREAM and Zorro. In: Advances in Cryptology – EUROCRYPT 2015. Ed. by E. Oswald and M. Fischlin. Vol. 9056. LNCS. Springer, 2015, pp. 254–283. DOI: 10.1007/978-3-662-46800-5_11 (pp. 3, 48, 49).

[LMS+15] M. Lamberger, F. Mendel, M. Schläffer, C. Rechberger, and V. Rijmen. The Rebound Attack and Subspace Distinguishers: Application to Whirlpool. In: Journal of Cryptology 28.2 (2015), pp. 257–296. DOI: 10.1007/s00145-013-9166-5 (pp. 47, 164).

[LP07] G. Leander and A. Poschmann. On the Classification of 4 Bit S-Boxes. In: Arithmetic of Finite Fields - WAIFI 2007. Ed. by C. Carlet and B. Sunar. Vol. 4547. LNCS. Springer, 2007, pp. 159–176. DOI: 10.1007/978-3-540-73074-3_13 (pp. 32, 77).

[LR88] M. Luby and C. Rackoff. How to Construct Pseudorandom Permutations from Pseudorandom Functions. In: SIAM Journal on Computing 17.2 (1988), pp. 373–386. DOI: 10.1137/0217022 (p. 271).

[LSWD04] T. V. Le, R. Sparr, R. Wernsdorf, and Y. Desmedt. Complementation-Like and Cyclic Properties of AES Round Functions. In: Advanced Encryption Standard - AES 2004. Ed. by H. Dobbertin, V. Rijmen, and A. Sowa. Vol. 3373. LNCS. Springer, 2004, pp. 128–141. DOI: 10.1007/11506447_11 (p. 59).

[LTW18] G. Leander, C. Tezcan, and F. Wiemer. Searching for Subspace Trails and Truncated Differentials. In: IACR Transactions on Symmetric Cryptology 2018.1 (2018), pp. 74–100. DOI: 10.13154/tosc.v2018.i1.74-100. URL: https://doi.org/10.13154/tosc.v2018.i1.74-100 (p. 50).

[Luc01] S. Lucks. The Saturation Attack - A Bait for Twofish. In: Fast Software Encryption – FSE 2001. Ed. by M. Matsui. Vol. 2355. LNCS. Springer, 2001, pp. 1–15. DOI: 10.1007/3-540-45473-X_1 (p. 34).

[LW17] C. Li and Q. Wang. Design of Lightweight Linear Diffusion Layers from Near-MDS Matrices. In: IACR Trans. Symmetric Cryptol. 2017.1 (2017), pp. 129–155. DOI: 10.13154/tosc.v2017.i1.129-155 (p. 245).

[Mat93] M. Matsui. Linear Cryptanalysis Method for DES Cipher. In: Advances in Cryptology – EUROCRYPT 1993. Ed. by T. Helleseth. Vol. 765. LNCS. Springer, 1993, pp. 386–397. DOI: 10.1007/3-540-48285-7_33 (pp. 2, 30, 96, 203, 251).

[Mat94] M. Matsui. The First Experimental Cryptanalysis of the Data Encryption Standard. In: Advances in Cryptology – CRYPTO 1994. Ed. by Y. Desmedt. Vol. 839. LNCS. Springer, 1994, pp. 1–11. DOI: 10.1007/3-540-48658-5_1 (pp. 2, 30, 96).

[Mat97] M. Matsui. New Block Encryption Algorithm MISTY. In: Fast Software Encryption – FSE 1997. Ed. by E. Biham. Vol. 1267. LNCS. Springer, 1997, pp. 54–68. DOI: 10.1007/BFb0052334 (p. 2).

[MDRM10] H. Mala, M. Dakhilalian, V. Rijmen, and M. Modarres-Hashemi. Improved Impossible Differential Cryptanalysis of 7-Round AES-128. In: Progress in Cryptology – INDOCRYPT 2010. Ed. by G. Gong and K. C. Gupta. Vol. 6498. LNCS. Springer, 2010, pp. 282–291. DOI: 10.1007/978-3-642-17401-8_20 (pp. 35, 38).

[MJSC16] P. Méaux, A. Journault, F. Standaert, and C. Carlet. Towards Stream Ciphers for Efficient FHE with Low-Noise Ciphertexts. In: Advances in Cryptology – EUROCRYPT 2016. Ed. by M. Fischlin and J. Coron. Vol. 9665. LNCS. Springer, 2016, pp. 311–343. DOI: 10.1007/978-3-662-49890-3_13 (p. 198).

[MN17] B. Mennink and S. Neves. Optimal PRFs from Blockcipher Designs. In: IACR Trans. Symmetric Cryptol. 2017.3 (2017), pp. 228–252. DOI: 10.13154/tosc.v2017.i3.228-252 (pp. 3, 193).

[MOV96] A. Menezes, P. C. van Oorschot, and S. A. Vanstone. Handbook of Applied Cryptography. CRC Press, 1996 (p. 205).

[MP15] B. Mennink and B. Preneel. On the Impact of Known-Key Attacks on Hash Functions. In: Advances in Cryptology – ASIACRYPT 2015. Ed. by T. Iwata and J. H. Cheon. Vol. 9453. LNCS. Springer, 2015, pp. 59–84. DOI: 10.1007/978-3-662-48800-3_3 (pp. 158, 162).

[MR02] S. Murphy and M. J. B. Robshaw. Essential Algebraic Structure within the AES. In: Advances in Cryptology – CRYPTO 2002. Ed. by M. Yung. Vol. 2442. LNCS. Springer, 2002, pp. 1–16 (p. 47).

[MRST09] F. Mendel, C. Rechberger, M. Schläffer, and S. S. Thomsen. The Rebound Attack: Cryptanalysis of Reduced Whirlpool and Grøstl. In: Fast Software Encryption – FSE 2009. Ed. by O. Dunkelman. Vol. 5665. LNCS. Springer, 2009, pp. 260–276. DOI: 10.1007/978-3-642-03317-9_16 (pp. 47, 164).

[MS78] F. MacWilliams and N. Sloane. The Theory of Error-Correcting Codes. 2nd. North-holland Publishing Company, 1978 (pp. 32, 245).

[Mur11] S. Murphy. The Return of the Cryptographic Boomerang. In: IEEE Trans. Information Theory 57.4 (2011), pp. 2517–2521. DOI: 10.1109/TIT.2011.2111091 (p. 109).

[MVO96] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot. Handbook of Applied Cryptography. CRC Press, Inc., 1996 (p. 9).

[Nik10] I. Nikolic. Tweaking AES. In: Selected Areas in Cryptography – SAC 2010. Ed. by A. Biryukov, G. Gong, and D. R. Stinson. Vol. 6544. LNCS. Springer, 2010, pp. 198–210. DOI: 10.1007/978-3-642-19574-7_14 (p. 59).

[NK92] K. Nyberg and L. R. Knudsen. Provable Security Against Differential Cryptanalysis. In: Advances in Cryptology – CRYPTO 1992. Ed. by E. F. Brickell. Vol. 740. LNCS. Springer, 1992, pp. 566–574. DOI: 10.1007/3-540-48071-4_41 (pp. 28, 225, 250).

[NK95] K. Nyberg and L. R. Knudsen. Provable Security Against a Differential Attack. In: Journal of Cryptology 8.1 (1995), pp. 27–37. DOI: 10.1007/BF00204800 (pp. 5, 28, 197).

[Nyb91] K. Nyberg. Perfect Nonlinear S-Boxes. In: Advances in Cryptology – EUROCRYPT 1991. Ed. by D. W. Davies. Vol. 547. LNCS. Springer, 1991, pp. 378–386. DOI: 10.1007/3-540-46416-6_32 (p. 77).

[Nyb94] K. Nyberg. S-boxes and Round Functions with Controllable Linearity and Differential Uniformity. In: Fast Software Encryption - FSE 1994. Ed. by B. Preneel. Vol. 1008. LNCS. Springer, 1994, pp. 111–130. DOI: 10.1007/3-540-60590-8_9 (pp. 28, 203).

[Nyb96] K. Nyberg. Generalized Feistel Networks. In: Advances in Cryptology – ASIACRYPT 1996. Ed. by K. Kim and T. Matsumoto. Vol. 1163. LNCS. Springer, 1996, pp. 91–104. DOI: 10.1007/BFb0034838 (pp. 6, 211, 213).

[OCo93] L. O'Connor. On the Distribution of Characteristics in Bijective Mappings. In: Advances in Cryptology – EUROCRYPT 1993. Ed. by T. Helleseth. Vol. 765. LNCS. Springer, 1993, pp. 360–370. DOI: 10.1007/3-540-48285-7_31 (p. 135).

[OCo94] L. O'Connor. Properties of Linear Approximation Tables. In: Fast Software Encryption - FSE 1994. Ed. by B. Preneel. Vol. 1008. LNCS. Springer, 1994, pp. 131–136. DOI: 10.1007/3-540-60590-8_10 (p. 136).

[Pat03] J. Patarin. Luby-Rackoff: 7 Rounds Are Enough for $2^{\mathrm{n(1\text{-}epsilon)}}$Security. In: CRYPTO 2003. Ed. by D. Boneh. Vol. 2729. LNCS. Springer, 2003, pp. 513–529. DOI: 10.1007/978-3-540-45146-4_30 (p. 271).

[Pat04] J. Patarin. Security of Random Feistel Schemes with 5 or More Rounds. In: CRYPTO 2004. Ed. by M. K. Franklin. Vol. 3152. LNCS. Springer, 2004, pp. 106–122. DOI: 10.1007/978-3-540-28628-8_7 (p. 271).

[Pha04] R. C. Phan. Impossible differential cryptanalysis of 7-round Advanced Encryption Standard (AES). In: Inf. Process. Lett. 91.1 (2004), pp. 33–38. DOI: 10.1016/j.ipl.2004.02.018 (pp. 3, 37).

[PSC+02] S. Park, S. H. Sung, S. Chee, E. Yoon, and J. Lim. On the Security of Rijndael-Like Structures against Differential and Linear Cryptanalysis. In: Advances in Cryptology – ASIACRYPT 2002. Ed. by Y. Zheng. Vol. 2501. LNCS. Springer, 2002, pp. 176–191. DOI: 10.1007/3-540-36178-2_11 (p. 33).

[PSLL03] S. Park, S. H. Sung, S. Lee, and J. Lim. Improving the Upper Bound on the Maximum Differential and the Maximum Linear Hull Probability for SPN Structures and AES. In: Fast Software Encryption – FSE 2003. Ed. by T. Johansson. Vol. 2887. LNCS. Springer, 2003, pp. 247–260. DOI: 10.1007/978-3-540-39887-5_19 (p. 33).

[PSSW09] B. Pinkas, T. Schneider, N. P. Smart, and S. C. Williams. Secure Two-Party Computation Is Practical. In: Advances in Cryptology – ASIACRYPT 2009. Ed. by M. Matsui. Vol. 5912. LNCS. Springer, 2009, pp. 250–267. DOI: 10.1007/978-3-642-10366-7_15 (p. 16).

[QSLG17] K. Qiao, L. Song, M. Liu, and J. Guo. New Collision Attacks on Round-Reduced Keccak. In: Advances in Cryptology - EUROCRYPT 2017. Ed. by J. Coron and J. B. Nielsen. Vol. 10212. LNCS. 2017, pp. 216–243. DOI: 10.1007/978-3-319-56617-7_8 (p. 217).

[RBH17] S. Rønjom, N. G. Bardeh, and T. Helleseth. Yoyo Tricks with AES. In: Advances in Cryptology – ASIACRYPT 2017. Ed. by T. Takagi and T. Peyrin. Vol. 10624. LNCS. Springer, 2017, pp. 217–243. DOI: 10.1007/978-3-319-70694-8_8 (pp. 34, 35, 44, 109, 115).

[RDP+96] V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers, and E. D. Win. The Cipher SHARK. In: Fast Software Encryption – FSE 1996. Ed. by D. Gollmann. Vol. 1039. LNCS. Springer, 1996, pp. 99–111. DOI: 10.1007/3-540-60865-6_47 (pp. 6, 239, 244).

[RSS17] D. Rotaru, N. P. Smart, and M. Stam. Modes of Operation Suitable for Computing on Encrypted Data. In: IACR Trans. Symmetric Cryptol. 2017.3 (2017), pp. 294–324. DOI: 10.13154/tosc.v2017.i3.294-324 (p. 210).

[RST18] C. Rechberger, H. Soleimany, and T. Tiessen. Cryptanalysis of Low-Data Instances of Full LowMCv2. In: IACR Trans. Symmetric Cryptol. 2018.3 (2018), pp. 163–181. DOI: 10.13154/tosc.v2018.i3.163-181. URL: https://doi.org/10.13154/tosc.v2018.i3.163-181 (p. 241).

[SB02] A. A. Selçuk and A. Biçak. On Probability of Success in Linear and Differential Cryptanalysis. In: Security in Communication Networks - SCN 2002. Ed. by S. Cimato, C. Galdi, and G. Persiano. Vol. 2576. LNCS. Springer, 2002, pp. 174–185. DOI: 10.1007/3-540-36413-7_13 (pp. 30, 96).

[Sel08] A. A. Selçuk. On Probability of Success in Linear and Differential Cryptanalysis. In: Journal of Cryptology 21.1 (2008), pp. 131–147. DOI: 10.1007/s00145-007-9013-7 (pp. 30, 96).

[Sha49] C. E. Shannon. Communication theory of secrecy systems. In: Bell System Technical Journal 28.4 (1949), pp. 656–715 (pp. 10, 11).

[Sho99] P. W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. In: SIAM Review 41.2 (1999), pp. 303–332. DOI: 10.1137/S0036144598347011 (p. 17).

[SLG+16] B. Sun, M. Liu, J. Guo, L. Qu, and V. Rijmen. New Insights on AES-Like SPN Ciphers. In: Advances in Cryptology – CRYPTO 2016. Ed. by M. Robshaw and J. Katz. Vol. 9814. LNCS. Springer, 2016, pp. 605–624. DOI: 10.1007/978-3-662-53018-4_22 (pp. 136, 137, 139).

[SLQL10] B. Sun, R. Li, L. Qu, and C. Li. SQUARE attack on block ciphers with low algebraic degree. In: SCIENCE CHINA Information Sciences 53.10 (2010), pp. 1988–1995. DOI: 10.1007/s11432-010-4061-2 (p. 43).

[SLR+15] B. Sun, Z. Liu, V. Rijmen, R. Li, L. Cheng, Q. Wang, H. AlKhzaimi, and C. Li. Links Among Impossible Differential, Integral and Zero Correlation Linear Cryptanalysis. In: Advances in Cryptology – CRYPTO 2015. Ed. by R. Gennaro and M. Robshaw. Vol. 9215. LNCS. Springer, 2015, pp. 95–115. DOI: 10.1007/978-3-662-47989-6_5 (p. 43).

[SM10] T. Suzaki and K. Minematsu. Improving the Generalized Feistel. In: Fast Software Encryption – FSE 2010. Ed. by S. Hong and T. Iwata. Vol. 6147. LNCS. Springer, 2010, pp. 19–39. DOI: 10.1007/978-3-642-13858-4_2 (p. 214).

[SMMK12] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi. TWINE: A Lightweight Block Cipher for Multiple Platforms. In: Selected Areas in Cryptography – SAC 2012. Ed. by L. R. Knudsen and H. Wu. Vol. 7707. LNCS. Springer, 2012, pp. 339–354. DOI: 10.1007/978-3-642-35999-6_22 (p. 214).

[SY11] Y. Sasaki and K. Yasuda. Known-Key Distinguishers on 11-Round Feistel and Collision Attacks on Its Hashing Modes. In: Fast Software Encryption – FSE 2011. Ed. by A. Joux. Vol. 6733. LNCS. Springer, 2011, pp. 397–415. DOI: 10.1007/978-3-642-21702-9_23 (p. 157).

[Tie16a] T. Tiessen. Polytopic Cryptanalysis. In: Advances in Cryptology – EUROCRYPT 2016. Ed. by M. Fischlin and J. Coron. Vol. 9665. LNCS. Springer, 2016, pp. 214–239. DOI: 10.1007/978-3-662-49890-3_9 (pp. 35, 45, 46, 109, 114).

[Tie16b] T. Tiessen. Secure Block Ciphers - Cryptanalysis and Design. PhD Thesis. Technical University of Denmark. 2016 (p. 9).

[TKKL15] T. Tiessen, L. R. Knudsen, S. Kölbl, and M. M. Lauridsen. Security of the AES with a Secret S-Box. In: Fast Software Encryption – FSE 2015. Ed. by G. Leander. Vol. 9054. LNCS. Springer, 2015, pp. 175–189. DOI: 10.1007/978-3-662-48116-5_9 (pp. 4, 136, 146, 193).

[Tod15a] Y. Todo. Integral Cryptanalysis on Full MISTY1. In: Advances in Cryptology – CRYPTO 2015. Ed. by R. Gennaro and M. Robshaw. Vol. 9215. LNCS. Springer, 2015, pp. 413–432. DOI: 10.1007/978-3-662-47989-6_20 (p. 2).

[Tod15b] Y. Todo. Structural Evaluation by Generalized Integral Property. In: Advances in Cryptology – EUROCRYPT 2015. Ed. by E. Oswald and M. Fischlin. Vol. 9056. LNCS. Springer, 2015, pp. 287–314. DOI: 10.1007/978-3-662-46800-5_12 (pp. 2, 42, 223).

[Tod17] Y. Todo. Integral Cryptanalysis on Full MISTY1. In: Journal of Cryptology 30.3 (2017), pp. 920–959. DOI: 10.1007/s00145-016-9240-x (p. 2).

[Tun12] M. Tunstall. Improved ”Partial Sums”-based Square Attack on AES. In: International Conference on Security and Cryptography - SECRYPT 2012. Ed. by P. Samarati, W. Lou, and J. Zhou. SciTePress, 2012, pp. 25–34 (p. 35).

[Unr15] D. Unruh. Non-Interactive Zero-Knowledge Proofs in the Quantum Random Oracle Model. In: Advances in Cryptology – EUROCRYPT 2015. Ed. by E. Oswald and M. Fischlin. Vol. 9057. LNCS. Springer, 2015, pp. 755–784. DOI: 10.1007/978-3-662-46803-6_25 (p. 18).

[Val08] P. Valiant. Incrementally Verifiable Computation or Proofs of Knowledge Imply Time/Space Efficiency. In: Theory of Cryptography - TCC 2008. Ed. by R. Canetti. Vol. 4948. LNCS. Springer, 2008, pp. 1–18. DOI: 10.1007/978-3-540-78524-8_1 (p. 210).

[Vau94] S. Vaudenay. On the Need for Multipermutations: Cryptanalysis of MD4 and SAFER. In: Fast Software Encryption - FSE 1994. Ed. by B. Preneel. Vol. 1008. LNCS. Springer, 1994, pp. 286–297. DOI: 10.1007/3-540-60590-8_22 (p. 32).

[Vau96] S. Vaudenay. On the Weak Keys of Blowfish. In: Fast Software Encryption – FSE 1996. Ed. by D. Gollmann. Vol. 1039. LNCS. Springer, 1996, pp. 27–32. DOI: 10.1007/3-540-60865-6_39 (p. 135).

[Wag99] D. A. Wagner. The Boomerang Attack. In: Fast Software Encryption - FSE 1999. Ed. by L. R. Knudsen. Vol. 1636. LNCS. Springer, 1999, pp. 156–170. DOI: 10.1007/3-540-48519-8_12 (pp. 43, 253).

[WCC+16] M. Wang, T. Cui, H. Chen, L. Sun, L. Wen, and A. Bogdanov. Integrals Go Statistical: Cryptanalysis of Full Skipjack Variants. In: Fast Software Encryption – FSE 2016. Ed. by T. Peyrin. Vol. 9783. LNCS. Springer, 2016, pp. 399–415. DOI: 10.1007/978-3-662-52993-5_20 (p. 170).

[WGR18] Q. Wang, L. Grassi, and C. Rechberger. Zero-Sum Partitions of PHOTON Permutations. In: Topics in Cryptology - CT-RSA 2018. Ed. by N. P. Smart. Vol. 10808. LNCS. Springer, 2018, pp. 279–299. DOI: 10.1007/978-3-319-76953-0_15 (p. 224).

[WPS+12] L. Wei, T. Peyrin, P. Sokolowski, S. Ling, J. Pieprzyk, and H. Wang. On the (In)Security of IDEA in Various Hashing Modes. In: Fast Software Encryption – FSE 2012. Ed. by A. Canteaut. Vol. 7549. LNCS. Springer, 2012, pp. 163–179. DOI: 10.1007/978-3-642-34047-5_10 (p. 157).

[WWGY14] Y. Wang, W. Wu, Z. Guo, and X. Yu. Differential Cryptanalysis and Linear Distinguisher of Full-Round Zorro. In: ACNS 2014. Ed. by I. Boureanu, P. Owesarski, and S. Vaudenay. Vol. 8479. LNCS. Springer, 2014, pp. 308–323. DOI: 10.1007/978-3-319-07536-5_19 (p. 240).

[Yao86] A. C. Yao. How to Generate and Exchange Secrets (Extended Abstract). In: 27th Annual Symposium on Foundations of Computer Science - October 1986. IEEE Computer Society, 1986, pp. 162–167. DOI: 10.1109/SFCS.1986.25 (p. 17).

[ZBL+15] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede. RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. In: SCIENCE CHINA Information Sciences 58.12 (2015), pp. 1–15. DOI: 10.1007/s11432-015-5459-7 (pp. 98, 99).

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.