Sebastian Ramacher

Cryptographic Schemes with Enhanced Security Properties and Post-Quantum Instantiations

DOCTORAL THESIS

to achieve the university degree of

Doktor der technischen Wissenschaften

submitted to

Graz University of Technology

Supervisor

Prof. Christian Rechberger

Assessor

Prof. Tibor Jager

Institute of Applied Information Processing and Communications

Graz, April 2019

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the dclard sources/resources, and that I have explicitely 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.

Sebastian Ramacher
Graz, April 2019

Abstract

Cryptography is a key building block for today's interconnected world. Standard primitives like encryption schemes, signature schemes and key exchanges find everyday use in major protocols building the backbone of secure communication. Hence it is of great importance to thoroughly analyze cryptographic schemes and protocols. Modern cryptography provides us with the necessary methods and tools to model the desired security properties and to then also prove particular constructions secure. Besides classical schemes and their security guarantees, this toolbox also allows us to precisely define enhanced security properties to cover adversaries gaining new attack possibilities or to define new features useful for applications.

We study various of these enhanced security properties: First, we enhance the security notions of proxy re-encryption with forward secrecy. This security property is of particular importance since secret keys might be leaked due to attacks on users or servers or inadvertently due to human error. In the setting of proxy re-encryption, where the proxy is always online to perform its task, reducing the damages of a breach on the proxy's side improves the overall trust in the system. Second, we extend classical signature schemes in the discrete logarithm setting with double-signature extraction in a black-box way. Thereby we obtain signature schemes which penalize dishonest behavior by leaking the secret key in this case. Hence signers are incentivized to behave honestly and to not sign contradicting statements. For all these schemes we provide efficient instantiations and prove them secure.

Additionally, we are concerned with the security of our constructions against adversaries with access to efficient quantum computers. For post-quantum secure instantiations of our schemes we base the constructions on symmetric-key primitives. Their security—even when considering quantum attacks—is relatively well understood. On the way to constructing post-quantum secure ring signatures and double-authentication-preventing signatures, we improve non-interactive zero-knowledge proofs for arithmetic circuits. For this proof system, we analyze and select compatible one-way functions with low multiplicative complexity. By proving statements with respect to the one-way function, we also obtain an efficient signature scheme solely relying on symmetric-key primitives.

Acknowledgements

Completion of a PhD thesis is a large project and requires the support from many people. First of all, I want to thank my girlfriend Theresa, my family and friends who accompanied my on this journey. I am very grateful for the support and encouragement from my colleagues and mentors David Derler and Daniel Slamanig. Both David and Daniel provided guidance and, most importantly, provided an environment where ideas could flourish and develop into interesting papers. From the very beginning of my PhD studies until the end they have been a constant source of inspiration.

I want to express my sincere gratitude to my advisor Christian Rechberger for the continuous support during my PhD studies and also for the change to delve into aspects of symmetric-key cryptography. He also provided me the opportunity to teach classes on IT security and modern public-key cryptography. I greatly enjoyed teaching these classes over the last years. I also want to thank Tibor Jager for serving as external assessor.

Furthermore, I want to especially thank Daniel Kales for interesting discussions on the implementation aspects of Picnic, as well as discussions and collaborations on fruitful ideas and topics. I also want to thank Greg Zaverucha for the opportunity to visit Microsoft Research and the continued collaboration on Picnic. Finally, I want to thank my colleagues, students and co-authors: Andreas Abraham, Martin R. Albrecht, Dieter and Harald Bratko, Melissa Chase, Itai Dinur, Christoph Dobraunig, Maria Eichlseder, Alfred Geroldinger, Steven Goldfeder, Alexander Grass, Lorenzo Grassi, Simon Guggi, Christian Hanser, Lukas Helminger, Felix Hörandner, Léo Perrin, Angela Promitzer, Stephan Krenn, Marin Krmpotić, Thomas Lorünser, Reinhard Lüftenegger, Olamide Omolola, Claudio Orlandi, Andreas Reinhart, Dragos Rotaru, Simon Roth, Arnab Roy, Markus Schofnegger, Dominik Schreilechner, and Christoph Striecks.

Sebastian Ramacher
Graz, April 2019

Table of Contents

Part I.

Background

1 Introduction

Since the internet's inception, its use has grown rapidly over the last years [Sta19; Ste18]. With this increase, every day use of computers and computing in general has observed a paradigm shift. Nowadays, documents are edited online in services such as Google Docs or Microsoft Office 365 [Pol15], pictures and videos are shared with all internet users using, e.g. Instagram or Youtube, or they are shared more selectively using cloud storage solutions such as Dropbox, Google Drive or Microsoft OneDrive. But also computing tasks are outsourced to cloud providers that offer access to machines in data centres around the globe. These tasks include general purpose computing or hosting of simple websites to online stores, but also data analysis and machine learning tasks that profit from dedicated hardware [Sae18]. This new paradigm however implies a shift of trust. As we move data to the cloud, we place a significant amount of trust in the service providers. After all, they get access to potentially confidential documents, pictures, or videos stored on their servers.

The recent push [Sch18] to encrypt all communication on the internet, driven by the major web browser vendors and cloud service providers alike, ensures end-to-end security between users and services by using, e.g., Transport Layer Security (TLS) [Res18]. When considering the use-cases discussed above, the picture gets more complicated. Simply encrypting data before uploading it to a cloud service is enough if we only consider the cloud as storage solution, but functionality that can otherwise be provided if the server gets to see plain data quickly becomes impossible to provide. Furthermore, relying on TLS to secure communication on the internet, also requires us to put significant trust in the currently deployed public key infrastructure (PKI) and certificate authorities (CAs). Attacks on CAs in recent years or misbehaving CAs, e.g., the incidents involving Comodo [Phi11] and DigiNotar [Adk11], have shown that this system is a potential weak point in our current security infrastructure. Therefore, new systems are needed to detect misbehaving or compromised CAs.

Altogether, those observations motivate the study of cryptographic schemes that provide more functionality and enhanced security properties than conventional digital signatures, public-key encryption or key encryption provide. Modern cryptography provides us with the language and tools to precisely define and study these additional properties, leading to new powerful cryptographic schemes. Some of these schemes include extensions of public-key encryption schemes such as proxy re-encryption [BBS98], attributed-based encryption, identity-based encryption [Sha84], puncturable encryption [GM15] and many more. Similarly, for digital signatures these extensions include proxy signatures, ring signatures [RST01], group signatures [CvH91], and blind signatures [Cha82] to name a few.

As instantiations and implementations of these schemes mature, they find more and more adoption in practice. Proxy re-encryption is useful as basis for end-to-end encrypted data-sharing and collaboration platforms such as BeSafe [BeS17] or NuCypher [Nuñ18]. Identity-based and broadcast encryption is used by Cloudflare's Geo Key Manager [Sul17] for secret key management. Also, cryptocurrencies employ schemes with sophisticated properties. For example, CryptoNote [Cry; vSMJ$^+$12], which puts a focus on privacy, uses traceable ring signatures [FS07] to obtain spender anonymity. Monero, a cryptocurrency with the same goals, also deployed similar techniques [Mon; Noe15; SAL$^+$17] until October 2018.

One has to keep in mind, that while these schemes are interesting for many applications, many of the practically efficient ones are built from assumptions in the factoring or discrete logarithm setting. However, ever since Shor published a polynomial-time quantum algorithm for factoring and computing discrete logarithms [Sho94], we know that a sufficiently powerful quantum computer is able to break all schemes and protocols used in practice today, as well as the schemes with extended security properties. This fact motivates the study of cryptographic schemes with post-quantum (PQ) security, i.e., security against an adversary having access to a quantum computer. While we currently do not know of the existence of a sufficiently powerful quantum computer, NIST announced the post-quantum cryptography project (PQC)[1] with the goal to evaluate and eventually standardize post-quantum secure digital signature schemes, public-key encryption schemes as well as key encapsulation mechanisms.

Luckily however, we know of enough assumptions that are believed to be secure also against an adversary having access to a quantum computer. In fact, we have a large zoo of assumptions from a diverse set of hard problems available as basis for post-quantum secure schemes. These set of problems include the learning from errors problem in lattices [Reg06], solving systems of multivariate polynomial equations over finite fields [MI88], decoding of syndromes of linear codes over finite fields [McE78; Nie86], or the Diffie-Hellman problem from supersingular isogenies [JF11]. Some of these assumptions are older than three decades, but compared to the performance achievable with elliptic curves or RSA based cryptosystems, they were too inefficient to be serious con-

[1] https://csrc.nist.gov/groups/ST/post-quantum-crypto/index.html

tenders. Alternatively, besides these problems based on mathematical structures, building schemes from collision-resistant hash functions and one-way functions becomes viable as well. Even though cryptosystems constructable from these primitives are limited [IR89; Imp95; LM09], their well-understood resistance against quantum attacks when built from symmetric-key primitives turns them into an interesting choice for building for some post-quantum secure cryptosystems. In particular for digital signatures, schemes based on Lamport's one-time signature scheme such as XMSS [BDH11] have been proposed and also standardized [HBG$^+$18].

We are now in a position to discuss the goals of our thesis. First, however note, we follow the current practice in modern cryptography and underline the security of our constructions with security proofs. For security properties we introduce, this approach also requires us to precisely define them. This will then enable us to argue about the security of our schemes using reductionist security proofs.

Enhanced Security Properties of Cryptographic Schemes. The security of cryptosystems completely relies on the secrecy of the respective secret keys. For example, if for an encryption scheme a secret key is either accidentally leaked, or obtained by an adversary via another channel, e.g. by breaking into a server, the confidentiality of all the encrypted data under this key is immediately lost. Similarly, for signature schemes in the context of PKI or app stores and software updates, key leakage would enable others to sign valid certificates or software packages, respectively, and thus would break the trust in the current infrastructure. Examples of such incidents include Adobe publicly posting their secret keys [Mim17] or attackers being able to steal private keys to decrypt credit card information as it may have happened during the Marriot data breach [Fis18]. Hence, we want to investigate the security properties that handle key leakage.

A naïve mitigation strategy for secret-key leakages is to frequently change all involved keys of a system. For public-key cryptosystems that would however imply that new public keys have to be distributed in a secure way or that public keys are huge as the contain independent keys for all time periods. Hence, a system that could avoid these issues would be desirable. Consequently, Günther [Gün89] introduced the notion of forward security, or in the context of public-key encryption also forward secrecy. In a cryptosystem providing forward security, key leakage at some point in time does not weaken the security properties before the compromise. Additionally, they provide efficient non-interactive solutions that have fixed public keys, yet their size is (asymptotically) sublinear in the number of key switches. Forward security has been identified as an important security property of various different cryptographic primitives such as digital signatures [BM99], identification schemes [AAB$^+$02], public-key encryption [CHK03], and private-key cryptography [BY03]. More recently, forward security was also introduced in the context of asynchronous messaging [GM15] and zero round-trip time key exchanges [GHJ$^+$17; DJS$^+$18].

In contrast to forward secrecy ensuring the security of a cryptosystems even after key leakage, controlled key leakage opens new possibilities to enforce honest behavior. If for example in the case where a user misbehaves its keys are leaked and all the security guarantees are void, users can be incentivized to behave honestly. In particular, in the context of signatures such an enhanced scheme would prevent signers from signing contradicting statements. Especially in scenarios where trust in honestly behaving parties plays an important, e.g. the trust put CAs building the trust anchor of today's PKI, could profit from a cryptographic incentive to behave honestly. For CAs leaking their secret key would completely destroy their business model [BPS17]. This feature can be achieved by using a signature scheme including a security notion to extract the secret key on misuse, e.g. a signature scheme with double-signature extractability dubbed double-authentication-preventing signatures [PS14; PS17]. Combined with a system like Certificate Transparency [Lau14; LLK13] we could immediately detect and penalize misbehavior.

Post-Quantum Instantiations. As discussed above, we have hardness assumptions at our disposal, that are believed to be secure even in the case of quantum adversary. With the NIST PQC project, we see the first serious standardization efforts to prepare for an transition to post-quantum secure standard primitives. Throughout the last decade, various lines of work also investigated efficient constructions of schemes with enhanced security properties based on post-quantum secure assumptions. Examples of such schemes include ring signatures from linear codes [BM18], ring and group signatures from lattices [LLN+16], lattice-based proxy re-encryption [CCL+14; PRS+17], hierarchical identity-based encryption from lattices [ABB10; CHK+10].

When considering constructions that only rely on symmetric-key primitives, the possibilities for constructing schemes with enhanced properties are more limited [IR89; Imp95; LM09]. In this setting, i.e. in the world of "minicrypt" as defined by Impagliazzo, one-way functions and non-interactive zero-knowledge proofs are available, but public-key encryption cannot be obtained. Since group signatures with the standard security model imply public-key encryption [AW04; CG04], groups signatures are not obtainable. But we can ask if we can construct other signature schemes with a focus on anonymity such as ring signatures or group signatures in a weaker model.

In any case, symmetric-key primitives such as block ciphers enjoy relatively well-understood security properties even in the presence of a quantum adversary. Therefore, they are a natural choice to build one-way functions or collision-resistant hash functions and to then extend those to more advanced schemes. Although we have to account for Grover's algorithm [Gro96] for one-way functions and pre-image attacks, the concrete consequences on collision resistance are still under debate. A detailed analysis of the costs of a quantum attack, such as the one by Brassard et al. [BHT98], are worse than the best classical attacks [Ber09]. So symmetric-key primitives amount to a very conservative choice to build post-quantum secure scheme. Hence, signatures lifting the one-time signature schemes of Lamport [Lam79] or Winternitz [DSS05] have been optimized over the years and concluded in efficient variants such as SPHINCS [BHH+15] and XMSS [BDH11]. We are thus interested in building signature schemes with enhanced functionality, while still basing the security of the schemes on symmetric-key primitives.

1.1. Contributions

Our results can be clustered based on our two goals: first, we consider enhanced security properties of cryptographic schemes such as forward secrecy, double-authentication prevention and anonymous authentication, and second, we consider post-quantum secure signature schemes and variants obtained from symmetric-key primitives. In the remainder of this section, we give a high-level overview of our results. The context of our contributions is discussed in Chapter 2 and the technical discussion is postponed to Part II.

For our first goal, we investigate proxy re-encryption, which allows a semi-trusted proxy to re-encrypt ciphertexts encrypted for one public key to another public key with specially crafted re-encryption keys. In [DKL+18a], we introduce security notions for forward secrecy in the context of proxy re-encryption and provide the first forward-secure proxy re-encryption scheme. On the way to this construction, we also introduce forward-secure delegatable public-key encryption and combine it with linearly homomorphic encryption to obtain forward-secure proxy re-encryption. We also provide an alternative construction based on positively and negatively puncturable encryption.

Second, we consider controlled key-leakage as method to disincentivize dishonest behavior. In particular, we focus on double-authentication-preventing signatures, which leak the secret key when signing two contradicting statements, with the goal to extend conventional signature schemes to double-authentication-preventing signatures. In [DRS18c] we introduce weaker extraction notions tailored for the extraction of the secret key of the underlying signature scheme. We also provide a generic construction from discrete logarithm-based signature schemes equipped with a group homomorphism between the secret- and public-key space, which naturally exists in this setting. Therefore our construction applies to, e.g., ECDSA and Schnorr signatures.

Besides investigating the enhanced properties in the classical setting, we continue our work in the post-quantum setting. To have an efficient non-interactive zero-knowledge proof system available for more advanced schemes, we improve non-interactive zero-knowledge proofs of knowledge for arithmetic circuits in [CDG+17a]. The proof system combined with an one-way function optimized for low multiplicative complexity then gives rise to an efficient post-quantum secure digital signature scheme, where signature are proofs of knowledge of the pre-image of the one-way function. This line of work also builds the basis for the digital signature scheme Picnic [CDG+17b; CDG+19], which only relies on the security of symmetric-key primitives. It was selected as candidate for the second round of the NIST post-quantum cryptography project [AAA+19].

With an efficient proof system at hand, we investigate non-interactive zero-knowledge membership proofs for Merkle tree accumulators in [DRS18b]. Based on this accumulator, we then construct logarithm-sized ring signature scheme, which allows a signer to form an ad-hoc ring of signers identified by their public keys, whereas the signer creates signatures on behalf of the rings while staying anonymous. The underlying technique to construct ring signatures from accumulators additionally requires one-way functions mapping into the domain of the accumulator, hence Picnic combined with Merkle tree accumulators then yield a ring signature scheme.

Finally, we use the power of the proof system to lift our double-authentication-preventing signature construction to the post-quantum setting [DRS18a]. We obtain a generic compiler that extends any signature scheme with a one-way function mapping secret keys to public key combined with a compatible pseudo-random function family into a double-authentication-preventing signature scheme. Thereby we can avoid the limitations observed for our discrete logarithm-based construction and the construction is no longer limited by the size of the address space. As with our ring signature scheme, we discuss concrete instantiations by applying the compiler to Picnic.

1.2. Other Contributions

We now briefly discuss additional contributions not included in our thesis. Our work on homomorphic proxy-re authenticators and privacy aspects in Certificate Transparency fit to the goal of schemes with enhanced security properties. The contributions to very efficient implementations of LowMC and the design of suitable block ciphers are of particular importance for the performance of our post-quantum secure schemes. We make the author's contributions explicit for papers where the author did not contribute as one of the main authors. We partly borrow formulations from the abstracts/introductions of the referenced papers.

Homomorphic Proxy Re-Authenticators [DRS17]. We investigate a scenario where sensors submit their data to an aggregator potentially running in the cloud, that evaluates a function on the data without revealing the actual data to the aggregator. Additionally, after the aggregation the user can verify that the computations has been performed correctly. To that effect, we introduce homomorphic proxy re-authenticators. Our framework tackles multi-user data aggregation in a dynamic setting. We thereby consider independent keys of the single parties, the verifiability of the evaluation of general functions on the authenticated inputs by the sources, as well as privacy with respect to the aggregator.

As a means to achieve the strong privacy requirements imposed by our security model, we formally define the notion of homomorphic proxy re-encryption. Additionally, we present two modular constructions of proxy re-authenticator schemes for the class of linear functions, which differ regarding the strength of the provided privacy guarantees. On our way, we establish various novel building blocks: Firstly, we present a linearly homomorphic message authentication code which is suitable to be used in our construction. Secondly, to achieve the stronger privacy guarantees, we construct a homomorphic proxy re-encryption scheme for linear functions. All our proofs are modular in the sense that we separately prove the security of our building blocks. Our overall proofs then build upon the results obtained for the building blocks. Thus, our building blocks may as well easily be used in other constructions.

Privacy in Certificate Transparency [KOR19]. Public key infrastructure (PKI) based on certificate authorities is one of the cornerstones of secure communication over the internet. Certificates issued as part of this PKI provide authentication of web servers among others. Yet, the PKI ecosystem is susceptible to certificate misissuance and misuse attacks. To prevent those attacks, Certificate Transparency (CT) facilitates auditing of issued certificates and detecting certificates issued without authorization. Users that want to verify inclusion of certificates on CT log servers contact the CT server directly to retrieve inclusion proofs. This direct contact with the log server creates a privacy problem since the users' browsing activities could be recorded by the log server owner.

We build on top of Lueks and Goldberg's approach [LG15] for privacy-preserving retrieval of inclusion proofs from CT log servers. To achieve privacy there, clients fetch inclusion proofs using a multi-server private information retrieval (PIR) protocol. We, however, present a more scalable design for logging a huge number of certificates, which allows us to include small static partial inclusion proofs in a Signed Certificate Timestamp (SCT), a server's certificate or as a TLS extension. The client can then check the inclusion based on the partial proof and by fetching the missing parts of the proof using a PIR-based approach.

Specifically, our goal is to tackle the privacy issue without any changes to the TLS server side to ease the possibility of a fast deployment. In our approach, we split the Merkle tree containing all certificates into multiple tiers of smaller Merkle trees where the trees at the bottom contain certificates. This split can, for example, be based on a parameterizable time interval or a maximum number of certificates. The sub-trees, respectively their roots, are then combined into the larger tree containing all certificates. This separation of the certificates into smaller sub-trees then allows us to embed membership proofs concerning the sub-trees in an extension field of the SCT or as an X.509v3 extension [CSF+08] into the certificate itself. As the height of the larger tree is now considerably smaller than a single tree containing all certificates, the approach by Lueks et al. [LG15] using PIR to fetch the membership proofs, becomes practical again. Additionally, we use a different two-server PIR solution and make use of the work on distributed point functions by Gilboa et al. [GI14] to build an efficient two-party computationally secure PIR system and present a highly performant implementation.

Optimization of the Linear Layer of LowMC [DKP+19]. Our constructions for post-quantum secure signatures as well as our ring signature and double-authentication-preventing signature constructions use the block cipher LowMC [ARS+15; ARS+16] as building block. The number of XOR operations involved in LowMC is one of the limiting factors with respect to the performance of those schemes. As such, we revisit the open problem of the LowMC designers to reduce the complexity of its linear operations, focusing on instances with small partial non-linear layers.

The fact that the S-box only operates on parts of the state and all other operations are matrix multiplications gives use some freedom to re-arrange computations in the linear layer. In particular, it allows us to split and re-arrange round-key computations, constant additions, and linear layer operations in a way that the size of the involved matrices can be significantly reduced. We propose an alternative, yet equivalent description of LowMC with a new structure, effectively reducing the size of the LowMC instances used in Picnic by a multiplicative factor of 2.38x and 4.84x. Runtime-wise we obtain an improvement of a factor between 1.41x to 2.82x for LowMC encryption and by a factor between 1.34x to 2.01x for Picnic. We also evaluate the obtained improvements in the context of private set-intersection [HL08] based on garbled circuits [Yao86]. There we obtain runtime improvements of factors 12x to 24x.

On the way to this result, we also address the question of whether the linear layer description we provide is optimal. We can prove that no further optimizations that reduce the linear layer size are possible without changing their functionality. We also consider the complexity of generating LowMC instances, assumings its linear layers are sampled at random. We devise a new, more efficient sampling algorithm, that is useful in applications requiring frequent instance generation, e.g. for the RASTA design strategy [DEG+18].

This work is a merge of [Din18] and [KPP+17]. The author is one of the main authors of the latter. The author contributed to the optimization of the linear layer operations, implementation in Picnic and the evaluation in [DKP+19].

Feistel Structures for MPC and More [AGP+19]. In this work, we explore construction strategies for constructions of symmetric-key primitives, which benefit secure multiparty computation (MPC) applications. We continue with an old design idea by Nyberg and Knudsen [NK95], in which the round function of a Feistel network is the mapping $x \mapsto x^3$. It has only been shown recently, that this idea in form of the block cipher MiMC can lead to efficient instantiations for succinct non-interactive arguments of knowledge (SNARKs) [AGR+16] and MPC protocols [GRR+16].

We adopt an old approach of symmetric cryptography, namely so-called *Generalized Feistel networks* generalizing the approach taken by the designers of DES. Thereby we obtain a generalized MiMC (GMiMC), which can cope with prime or binary fields, and many field elements at once. For MPC applications, previous works [GRR+16; RSS17] did not take into account how to optimize the number of multiplications for a higher number of blocks and treated pseudorandom functions (PRFs) as a black-box when extending to more inputs. This is where our constructions shines the most in the context of MPC: if one chooses to encrypt multiple shares at once we can amortize the number of multiplications per share resulting in a more efficient preprocessing phase. We consider our work to be beneficial when there is a large number of blocks to encrypt.

Besides MPC protocols, we also explore the applications of our design strategy in the context of SNARKs and Picnic-style signatures. For the latter, LowMC was considered so far to clearly be the best choice for small signatures and efficient runtime. Using MiMC resulted in 10 times larger and hence unpractical signatures. We show that the picture is more complex and GMiMC can be much more efficient than the original MiMC. Due to the flexibility of the design, GMiMC is also competitive with LowMC, slightly performing better in both runtime and signature size.

The author contributed to the evaluation of the cipher design in the context of Picnic and also as choice of symmetric-key primitive to construct ring signatures.

1.3. About The Thesis

This thesis is a cumulative thesis consisting of two parts. The first part of this thesis, Part I, serves as a high-level overview on the field of public-key cryptography focusing on digital schemes and public-key encryption as well as variants thereof. This overview puts the publications covered in our thesis in Part II into context. In particular, it also discusses some techniques that we have applied in our work. Consequently, this part does not contain any new scientific contributions.

Part I is structured as follows: First, we recall some notions and concepts of modern cryptography in Section 2.2. Second, we discuss public-key encryption including proxy re-encryption in Section 2.4. Finally, Section 2.5 covers digital signatures including ring signatures and double-authentication-preventing signatures.

Part II contains the scientific contributions of our thesis by appending the papers and detailing the author's contribution to each paper. Regarding the contribution, we want to point out the following statement of the American Mathematical Society:

> In most areas of mathematics, joint research is a sharing of ideas and skills that cannot be attributed to the individuals separately. The roles of researchers are seldom differentiated (in the way they are in laboratory sciences, for example). Determining which person contributed which ideas is often meaningless because the ideas grow from complex discussions among all partners. ([Ame04])

We think that this statement also applies to cryptography.

2 Background

In this chapter we present definitions, notions and selected construction to put the publications into context. Throughout this chapter we assume familiarity with basic cryptographic concepts and computational complexity theory. For excellent introductions we refer the reader to the books of Smart [Sma16], Katz and Lindell [KL14], and Goldreich [Gol01; Gol04; Gol08; Gol10]. If not otherwise cited differently, we base our definitions on standard literature such as [Kat10; KL14], and in particular, we follow Katz and Lindell [KL14] in our discussion of the aspects of modern cryptography. We keep the definitions close to those used in our publications and partly borrow their formulations.

2.1. Notation

We introduce some notation we will use throughout this chapter. For $m, n \in \mathbb{N}, m \leq n$, we let $[m, n] = \{m, \ldots, n\}$, i.e. all natural numbers in the interval between m and n, and $[n] = [1, n]$. For a natural number $n \in \mathbb{N}$, we denote the ring of integers modulus n as $\mathbb{Z}_n$. We let $\kappa \in \mathbb{N}$ be the security parameter. To sample from a set S uniformly at random, we write $x \overset{R}{\leftarrow} S$. For an algorithm A, let $y \leftarrow \mathsf{A}(1^\kappa, x)$ be the process of running A, on input 1^κ and x, with access to uniformly random coins and assigning the result to y. We assume that all algorithms take 1^κ as input and we will sometimes omit to make this explicit. For an probabilistic algorithm A, we make the random coins r explicit by writing $\mathsf{A}(1^\kappa, x; r)$. An algorithm A is probabilistic polynomial time (PPT) if its running time is polynomially bounded in κ. In this case we also say that A is efficient. Additionally, to values polynomially bounded in κ, we write $n \leq \mathsf{poly}(\kappa)$. We use calligraphic letters, e.g., $\mathcal{A}$, for the algorithms representing adversaries in the security games. A function $f \colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ is negligible if

$$\forall c \, \exists \kappa_0 \, \forall \kappa \geq \kappa_0 \colon f(\kappa) \leq \frac{1}{\kappa^c}.$$

2.2. Modern Cryptography

Modern cryptography follows a systematic approach to study, understand and argue about the security of cryptographic schemes: First, *formal definitions* are introduced to capture the desired security properties. These definitions cover the available algorithms and inputs of a scheme, state how a correctly functioning system should behave in terms of functionality, as well as security goals a scheme should achieve. For the latter, experiments with a challenger interacting with an adversary are defined. In these experiments, simply put, the adversary is given a problem instance set up by the challenger and the adversary than needs to solve this problem.

Second, when schemes cannot be proven unconditionally secure, *assumptions* provide a hook to analyse the security of a schemes. These assumptions cover the hardness of solving precisely stated problems. Here we understand hardness in the sense that no efficient adversary can solve the problem except with negligible probability.

Third and finally, constructions are argued to be secure by providing *proofs of security*. The goal of these proofs is, e.g., to reduce the security of a cryptographic scheme to an assumption or the security properties of another scheme. These reductionist proofs are performed by contradiction, i.e. we assume that an efficient algorithm exists to win the security game and then build an efficient reduction such that we obtain an efficient algorithm to solve a hard problem. Thus, instead of studying the security of each and every scheme, the security proofs enable us to turn our focus to studying whether the assumptions are justified. We want to note the similarity of the proof technique with reductions in complexity theory, e.g. of **NP**-hard problems. They highlight the roots of modern cryptography in complexity theory.

Throughout all our work we employ a technique called *sequence of games* (cf. [Sho04] for an overview) to obtain clear and easy to comprehend security proofs when direct direct reductions would otherwise be too complex. The basic idea of this technique is to change the behavior of the challenger that interacts with the adversary, i.e. one constructs a sequence of games starting from the original attack experiment as Game 0, and continuously adapts it until one reaches Game n, which allows us to easily argue about the winning probability of Game n. To argue that the changes between successive games are sound, we show that the probability of wining the individual games are negligibly close. Typical transitions between games include:

Transitions based on indistinguishability. Here detection of the change by the adversary would imply an efficient method to distinguish two indistinguishable distributions.

Transitions based on failure events. Here one argues that two games proceed identically unless a failure event occurs. With this type of argument we can bound the differente of winning the original and modified game by the probability of the failure event. Additionally, this allows us to handle the probability of the failure event on its own.

Transitions based on bridging steps. Here one re-states the computation of certain values in an equivalent way. The changes made to the game are purely conceptual.

After at most polynomially many transitions one obtains an upper bound on the winning the probability of the initial game, e.g., if S_i denotes the event of winning game G_i, then we have a bound $\Pr[S_0] \leq \Pr[S_1] \leq \cdots \leq \Pr[S_n]$. Consequently, if the probability of winning G_n is negligible, so is the probability of wining G_0.

Following this approach to argue about security, we obtain proofs of security relative to the definition being considered and the used assumptions. However, if the assumptions turn out to be false, the definitions do not match the abilities of an adversary, or if the guarantees required by applications are not covered, the proofs of security may be meaningless.[1] Although this approach does not necessarily imply security in real world applications, it helps to provide confidence in the security of the cryptographic schemes.

2.2.1. Hardness Assumptions

Security arguments often rely on assumptions, that are widely believed to be hard to solve, but unproven. In the modern paradigm, these assumptions have to be defined very precisely, i.e. the problem statements are formalized unambiguously with a clearly defined goal an adversary has to achieve. On the one hand, we can then analyze the hardness of the assumptions on their own. On the other hand, they also provide a level of abstraction helping cryptographers to focus on analyzing the security of schemes.

In the following we discuss some hardness assumptions that we require for our constructions in [DKL+18a; DRS18c]. We will often call these assumptions obtained from hard problems related to some mathematical structures *structured hardness assumptions.*

2.2.1.1. Prime-Order Groups

First we look at assumptions related to the discrete logarithm problem in cyclic groups. Note that we write all groups using multiplicative notation. We first define a group generator $\mathcal{G}$:

Definition 1. A group generation algorithm $\mathcal{G}$ is an algorithm that takes a security parameter κ and outputs a cyclic group description $(\mathbb{G}, q, g)$ of a group $\mathbb{G}$ of prime order q and generator g, i.e. $\mathbb{G} = \langle g \rangle$.

Definition 2 (DLP)**.** The discrete logarithm problem (DLP) assumption holds relative to $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^\kappa)$, if for all PPT adversaries $\mathcal{A}$, there is a negligible function ε such that

$$\Pr\left[\begin{array}{l} x \xleftarrow{R} \mathbb{Z}_q, \\ x^* \leftarrow \mathcal{A}\left(\mathbb{G}, q, g, g^x\right) \end{array} \; : \; x = x^* \right] \leq \varepsilon(\kappa).$$

[1] A recent example, where the security model does not match the abilities of the adversary, is an attack on the 4-way handshake of WPA2 [VP17]. The attack does not violate the formally proven security properties, yet it has catastrophic impact on Wi-Fi security.

Definition 3 (DDH)**.** The decisional Diffie-Hellman (DDH) assumptions holds relative to $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^\kappa)$, if for all PPT adversaries $\mathcal{A}$, there is a negligible function ε such that

$$\left| \Pr\left[\begin{array}{l} b \xleftarrow{R} \{0,1\}, (x,y,z) \xleftarrow{R} \mathbb{Z}_q^3, \\ b^* \leftarrow \mathcal{A}\left(\mathbb{G}, q, g, g^x, g^y, g^{bxy+(1-b)z}\right) \end{array} \ : \ b = b^* \right] - \frac{1}{2} \right| \leq \varepsilon(\kappa).$$

Definition 4 (DLIN)**.** The decision linear (DLIN) assumptions holds relative to $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^\kappa)$, if for all PPT adversaries $\mathcal{A}$, there is a negligible function ε such that

$$\left| \Pr\left[\begin{array}{l} (u,v) \xleftarrow{R} \mathbb{G}^2, (x,y,z) \xleftarrow{R} \mathbb{Z}_q^3, b \xleftarrow{R} \{0,1\}, \\ b^* \leftarrow \mathcal{A}\left(\mathbb{G}, q, g, u, v, , u^x, v^y, g^{b\cdot(x+y)+(1-b)z}\right) \end{array} \ : \ b = b^* \right] - \frac{1}{2} \right| \leq \varepsilon(\kappa).$$

Popular instantiations for groups where we assume that the hardness assumptions hold are elliptic curve groups or groups from other Abelian varities such as Edwards curves. Before the advance of elliptic curve based cryptography multiplicative subgroups of finite fields were a popular choice,[2] but nowadays they cannot compete with performance figures of elliptic curves. However, since the discrete logarithm problem can be transformed into a factorization problem, Shor's factorization algorithm [Sho94] can also be used to efficiently compute discrete logarithms on a powerful enough quantum computer. Therefore, the DLP and related assumptions do not yield post-quantum secure constructions.

2.2.1.2. Bilinear Groups

The second category of assumptions are based on bilinear groups. Before discussing assumptions related to bilinear groups, we shortly recall pairings. We refer to [EJ17] for an overview on all aspects of pairing-based cryptography.

Definition 5 (Bilinear pairing)**.** Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be groups and $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. The map e is called a bilinear pairing if

- e is bilinear, that is, for all $x, y \in \mathbb{G}_1$ and $u, v \in \mathbb{G}_2$ it holds that

$$e(x \cdot y, u) = e(x, u) \cdot e(y, u) \text{ and } e(x, u \cdot v) = e(x, u) \cdot e(x, v).$$

- e is non-degenerate, that is there exist non-trivial $g \in \mathbb{G}_1$ and $\hat{g} \in \mathbb{G}_2$ such that $e(g, \hat{g}) \neq 1$.
- e is efficiently computable, i.e there exists a polynomial time algorithm to compute e.

Bilinear pairings can be classified into different types which is based on the choice of $\mathbb{G}_1$ and $\mathbb{G}_2$:

Definition 6. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be a bilinear pairing.

[2] The group operation on elliptic curves is usually described as addition. Due to the origin of the DLP in multiplicative groups, we stick to multiplicative notation.

1. The pairing e is said to be of *Type 1* if $\mathbb{G}_1 = \mathbb{G}_2$.
2. The pairing e is said to be of *Type 2* if $\mathbb{G}_1 \neq \mathbb{G}_2$ and there exists an efficiently computable group isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$.
3. The pairing e is said to be of *Type 3* if $\mathbb{G}_1 \neq \mathbb{G}_2$ and no efficiently computable group isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ is known to exist.

Definition 7. A bilinear-group generation algorithm BG is a PPT algorithm that takes a security parameter κ and the type $t \in [3]$ and outputs a bilinear group description $(t, e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, \hat{g}, \mathbf{g})$ with $\mathbb{G}_1 = \langle g \rangle$, $\mathbb{G}_2 = \langle \hat{g} \rangle$ and $\mathbb{G}_T = \langle \mathbf{g} \rangle$, all three of prime order q and an asymmetric pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

We note that the DLIN assumption can be defined as in Definition 4 also for the case of a bilinear group generator. Since we require the bilinear decisional Diffie-Hellman assumption for Type 1 pairings in [DKL+18a], we recall it here. For an extensive framework for bilinear groups based hardness assumptions, we refer the reader to the “uber” assumption framework of Boyen [Boy08].

Definition 8 (BDDH)**.** The bilinear decisional Diffie-Hellman assumptions holds relative to $\mathsf{bg} \leftarrow \mathsf{BG}(1^\kappa, 1)$, if for all PPT adversaries $\mathcal{A}$, there is a negligible function ε such that

$$\Pr\left[\begin{array}{l} (r,s,t,u) \xleftarrow{R} \mathbb{Z}_q^4, b \xleftarrow{R} \{0,1\} \\ y \leftarrow \mathcal{A}\left(\mathsf{bg}, g^r, g^s, g^t, \mathbf{g}^{brst+(1-b)u}\right) \end{array} \; : \; b = b^* \right] \leq \varepsilon(\kappa)$$

The first theoretical constructions of bilinear pairings date back to 1940 when Weil introduced a pairing on Abelian varieties [Wei40], later followed by Tate's and Lichtenbaum's constructions [Tat57; Tat63; Lic69]. Only since the seminal work of Miller [Mil86; Mil04], we know of efficient algorithms to compute pairings. Pairings were first introduced into cryptography for breaking the DLP in elliptic curve groups [MVO91; FR94], and only later found applications in protocols such as Joux's three-party key-exchange protocol [Jou00], for providing the first construction of an efficient identity-based encryption scheme [BF01], and short signatures [BLS01].

Currently the most prominent choice for instantiating bilinear pairings are Type 3 pairings over Barreto-Naehrig (BN) curves [BN05] or Barreto-Lynn-Scott (BLS) curves [BLS02]. For both of these curves, $\mathbb{G}_1$ is an elliptic curve group over $\mathbb{F}_p$ with prime order q, $\mathbb{G}_2$ is again a group of the same order coming from a twist, and $\mathbb{G}_T$ is a subgroup of the multiplicative group of $\mathbb{F}_{p^{12}}$. Considering recent progress on solving the discrete logarithm problems in such fields, conservative estimations [BD17] recommend bit lengths for p where BLS curves and KSS curves [KSS08] are better choices than BN curves, as they provide more efficient arithmetic at these bit lengths.

We however note that, while Type 3 pairing implementations are the most efficient to date, designing schemes in the Type 1 setting might be easier. Fortunately, generic compilers [AHO16] exist to convert a Type 1 construction into one using Type 3 pairings.

2.2.2. Computational Models

Ideally, the hardness of breaking cryptographic schemes or protocols can be directly related to hardness of breaking a particular hardness assumption. If such reduction is possible we talk about a security reduction in the *standard model*. Proofs in the standard model are the most desirable ones, since than we do not have to fall back on idealized computational models for proving schemes secure. Sometimes, however, it is necessary to make further assumptions on the computation model, such as the common reference string model, the generic group model [Sho97; KM07], or the random oracle model (ROM). Besides the schemes in [DKL+18a] with security proofs in the standard model, all our other publications require the random oracle or the quantum-accessible variant. Hence we discuss the ROM in more detail.

In the ROM [BR93], hash functions are idealized by modeling them as oracles with return uniformly random responses. Upon receiving a query, the random oracle chooses a new uniform random value, stores it and answers. On receiving a query for a previously queried value, the RO answers with the stored value. In cases where where this behaviour of the random oracle is not enough for the security proof, programmability of the oracle [FLR+10], i.e. programming the oracle to return particular values upon particular queries, might be useful. While these values still need to be distributed in the same way, this technique for examples allows one to embed a challenge into the responses.

We want to note that there exist artificial schemes [CGH04] which can be proven secure in the ROM, but which are insecure when with any concrete hash function. Yet, as Koblitz and Menezes argue [KM15], a proof in the ROM does not indicate the presence of a real-world security weakness. Furthermore, the ROM has turned out to be a useful assumption for analysing efficient constructions. Alternatively, put in the words of Bellare and Rogaway: “Goals which are possible but impractical in the standard setting become practical in the random oracle setting.” [BR93]

The quantum-accessible random oracle model (QROM) [BDF+11] is a variant of the ROM, where an adversary can issue quantum queries to the RO. This approach allows to model an adversary that interacts with a classical challenger, or in other words, it handles the case where we are concerned with powerful adversaries attacking cryptosystems that are used on classical computers. Giving the adversary the possibility to perform quantum queries, however, means that the adversary may query the oracle in superposition, so the challenger also needs to return answers in superposition. Hence, some proof techniques, such as programmability of the RO or rewinding of the adversary [PS00], that work in the classical setting, do not carry over to the QROM or only work in certain scenarios. New techniques have been developed to prove security in the QROM, such as history-free reductions [BDF+11] and many others, e.g see [Zha12; BZ13; TU16; Unr17; KLS18; Zha18].

2.3. Cryptographic Building Blocks

Before we start discussing encryption and signature schemes, we first recall some of the basic building blocks that are used throughout our work. We often require one-way functions, pseudorandom functions, accumulators, Σ-protocols and non-interactive zero-knowledge proof systems.

2.3.1. One-Way Functions and Pseudorandom Function Families

We recall the definitions of one-way functions and pseudorandom function (families).

Definition 9 (OWF)**.** Let $f\colon S \to P$ be a function. For a PPT adversary $\mathcal{A}$ we define the advantage function as

$$\mathsf{Adv}^{\mathsf{OWF}}_{\mathcal{A},f}(\kappa) = \Pr\left[x \xleftarrow{R} S, x^* \leftarrow \mathcal{A}(1^\kappa, f(x)) : f(x) = f(x^*)\right].$$

The function f is one-way function (OWF) if it is efficiently computable and for all PPT adversaries $\mathcal{A}$ there exists a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{OWF}}_{\mathcal{A},f}(\kappa) \leq \varepsilon(\kappa).$$

Definition 10 (PRF)**.** Let $\mathcal{F}\colon \mathcal{S} \times D \to \mathsf{R}$ be a family of functions and let Γ be the set of all functions $D \to \mathsf{R}$. For a PPT distinguisher $\mathcal{D}$ we define the advantage function as

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{D},\mathcal{F}}(\kappa) = \left|\Pr\left[s \xleftarrow{R} \mathcal{S} : \mathcal{D}^{\mathcal{F}(s,\cdot)}(1^\kappa) = 1\right] - \Pr\left[f \xleftarrow{R} \Gamma : \mathcal{D}^{f(\cdot)}(1^\kappa) = 1\right]\right|.$$

$\mathcal{F}$ is a pseudorandom function (family) if it is efficiently computable and for all PPT distinguishers $\mathcal{D}$ there exists a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{D},\mathcal{F}}(\kappa) \leq \varepsilon(\kappa).$$

Our constructions in [CDG+17a; DRS18b; DRS18a] rely on one-way functions built from symmetric-key primitives. In particular, given a block cipher with encryption algorithm $\mathcal{E}(s, x)$, where s denotes the secret key and x the plaintext block, then we obtain a one-way function by fixing a random plaintext p and setting $f(x) = \mathcal{E}(x, p)$. Similarly, a pseudorandom function family can be constructed by using the folklore feed-forward construction $\mathcal{F}(s, x) = \mathcal{E}(s, x) \oplus x$ due to Davies and Meyer. For a recent discussion of PRF constructions from block ciphers we refer to [MN17].

More concretely, we rely on LowMC [ARS+15; ARS+16] as block cipher, which is a very parameterizable symmetric encryption scheme design enabling instantiation with low multiplicative depth and complexity. Given any block size, a choice for the number of S-boxes per round, and security expectations in terms of time and data complexity, instantiations can be found minimizing the multiplicative depth, the number of multiplications, or the number of multiplications per encrypted bit. A selection of parameters for LowMC is given in Table 1. All round numbers were generated using the parameter generation script[3] based on the round formulas by Rechberger et al. [RST18].

Block/key size ($n = k$)	S-boxes (m)	Data complexity (d)	Rounds (r)
128	10	1	20
192	10	1	30
256	10	1	38
128	10	128	32
192	10	192	45
256	10	256	58
128	1	1	182
192	1	1	284
256	1	1	363
128	1	128	287
192	1	192	413
256	1	256	537

Table 1: A selection of parameters for LowMC version 3 for 10 and 1 S-boxes, respectively. The parameter sets with $d = 1$ are of particular interest for Picnic.

Given the block size n, the number of S-boxes m per round, the key size k, and the number of rounds r, we first select uniformly at random round constants $C_i \overset{R}{\leftarrow} \mathbb{F}_2^n$ and regular matrices $L_i \overset{R}{\leftarrow} \mathbb{F}_2^{n \times n}$ for $i \in [r]$, as well as full rank matrices $K_i \overset{R}{\leftarrow} \mathbb{F}_2^{n \times k}$ for $i \in [0, r]$. Round keys are derived from the secret key using the respective key matrices K_i. The first key matrix K_0 is used for key whitening before the first round. Then LowMC applies multiple rounds composed of an S-box layer, a linear layer that multiplies the state with L_i, and finally adding the round constants and the round keys. Algorithm 1 gives a full description of the encryption algorithm, where SBOX is an m-fold parallel application of the same 3-bit S-box S on the first $3 \cdot m$ bits of the state. The 3-bit S-box is defined as

$$\begin{aligned} S \colon \mathbb{F}_2^3 &\to \mathbb{F}_2^3 \\ (a, b, c) &\mapsto (a + b \cdot c, a + b + a \cdot c, a + b + c + a \cdot b). \end{aligned}$$

For in-depth security analysis of this block cipher design we refer to [DLM+15; DEM15; RST18]. The latter also contains the latest formulas to derive round numbers for concrete instances of LowMC.

[3] https://github.com/lowmc/lowmc

Algorithm 1 LowMC encryption for key matrices $K_i \in \mathbb{F}_2^{n \times k}$ for $i \in [0, r]$, linear layer matrices $L_i \in \mathbb{F}_2^{n \times n}$ and round constants $C_i \in \mathbb{F}_2^n$ for $i \in [r]$.

```
Input: plaintext $p \in \mathbb{F}_2^n$ and key $y \in \mathbb{F}_2^k$
Output: ciphertext $s \in \mathbb{F}_2^n$
  $s \leftarrow K_0 \cdot y + p$
  for $i \in [r]$ do
      $s \leftarrow \mathrm{SBOX}(s)$
      $s \leftarrow L_i \cdot s$
      $s \leftarrow s + C_i + K_i \cdot y$
  end for
  return $s$
```

2.3.2. Accumulators

An accumulator [BdM93] allows one to accumulate a finite set $\mathcal{X}$ into a succinct value called the accumulator. For every element in the accumulated set, one can efficiently compute a witness certifying its membership. We recall the definition of static accumulators by Derler et al. [DHS15].

Definition 11 (Accumulator)**.** A static accumulator Λ is a tuple of PPT algorithms $(\mathsf{Gen}, \mathsf{Eval}, \mathsf{WitCreate}, \mathsf{Verify})$ which are defined as follows:

$\mathsf{Gen}(1^\kappa, t)$: This algorithm takes a security parameter κ and a parameter t. If $t \neq \infty$, then t is an upper bound on the number of elements to be accumulated. It returns a key pair $(\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda)$, where $\mathsf{sk}_\Lambda = \emptyset$ if no trapdoor exists. We assume that the accumulator public key pk_Λ implicitly defines the accumulation domain D_Λ.

$\mathsf{Eval}((\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda), \mathcal{X})$: This deterministic algorithm takes a key pair $(\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda)$ and a set $\mathcal{X}$ to be accumulated and returns an accumulator $\Lambda_\mathcal{X}$ together with some auxiliary information aux.

$\mathsf{WitCreate}((\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda), \Lambda_\mathcal{X}, \mathsf{aux}, x_i)$: This algorithm takes a key pair $(\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda)$, an accumulator $\Lambda_\mathcal{X}$, auxiliary information aux and a value x_i. It returns $\perp$, if $x_i \notin \mathcal{X}$, and a witness wit_{x_i} for x_i otherwise.

$\mathsf{Verify}(\mathsf{pk}_\Lambda, \Lambda_\mathcal{X}, \mathsf{wit}_{x_i}, x_i)$: This algorithm takes a public key pk_Λ, an accumulator $\Lambda_\mathcal{X}$, a witness wit_{x_i} and a value x_i. It returns 1 if wit_{x_i} is a witness for $x_i \in \mathcal{X}$ and 0 otherwise.

We require accumulators to be correct and collision free. For correctness, we require that for all $\kappa \in \mathbb{N}$, $(\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda) \leftarrow \mathsf{Gen}(1^\kappa, t)$, for a set $\mathcal{X}$ with $\|\mathcal{X}\| \leq t$, $\Lambda_\mathcal{X}, \mathsf{aux} \leftarrow \mathsf{Eval}((\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda), \mathcal{X})$, and for all $x \in \mathcal{X}$ we have that

$$\mathsf{Verify}\left(\mathsf{pk}_\Lambda, \Lambda_\mathcal{X}, \mathsf{WitCreate}\left((\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda), \Lambda_\mathcal{X}, \mathsf{aux}, x\right), x\right) = 1.$$

For collision-freeness we require that finding a witness for a non-accumulated value is hard.

Definition 12 (Collision-Freeness)**.** For an efficient adversary $\mathcal{A}$, we define the advantage function in the sense of collision-freeness as

$$\mathsf{Adv}^{\mathsf{CF}}_{\mathcal{A},\Lambda}(\kappa, t) = \Pr\left[\mathsf{Exp}^{\mathsf{CF}}_{\mathcal{A},\Lambda}(1^\kappa, t) = 1\right],$$

where the corresponding experiment is depicted in Experiment 1. If for any efficient adversary $\mathcal{A}$ there exists a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{CF}}_{\mathcal{A},\Lambda}(\kappa, t) \leq \varepsilon(\kappa),$$

then Λ is collision-free.

```
$\mathsf{Exp}^{\mathsf{CF}}_{\mathcal{A},\Lambda}(1^\kappa, t)$:
  $(\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda) \leftarrow \mathsf{Gen}(1^\kappa, t)$
  $(\mathsf{wit}^*_{x_i}, x^*_i, \mathcal{X}^*, r^*) \leftarrow \mathcal{A}^{\{\mathsf{E},\mathsf{W}\}}(\mathsf{pk}_\Lambda)$
    where oracle $\mathsf{E}(\mathcal{X}, r)$:
      return $\mathsf{Eval}((\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda), \mathcal{X}; r)$
    and oracle $\mathsf{W}(\Lambda_\mathcal{X}, \mathsf{aux}, x_i)$:
      return $\mathsf{WitCreate}((\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda), \Lambda_\mathcal{X}, \mathsf{aux}, x_i)$
  $\Lambda^* \leftarrow \mathsf{Eval}_{r^*}((\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda), \mathcal{X}^*)$
  return 1, if $\mathsf{Verify}(\mathsf{pk}_\Lambda, \Lambda^*, \mathsf{wit}^*_{x_i}, x^*_i) = 1 \ \wedge x^*_i \notin \mathcal{X}^*$
  return 0
```

Experiment 1: Collision-freeness experiment.

The definition of collision-freeness covers both randomized and deterministic accumulators. When considering deterministic accumulators, e.g. Merkle trees [Mer89], then r^* and the additional input r to the Eval oracle E can simply be omitted.

2.3.3. Σ-protocols

Let $L \subseteq \mathsf{X}$ be an **NP**-language with associated witness relation R so that

$$L = \{x \in \mathsf{X} \mid \exists w \colon R(x, w) = 1\}.$$

A Σ-protocol for language L is an interactive three move protocol between a prover and a verifier, where the prover proves knowledge of a witness w to the statement $x \in L$. For a language L, they are defined as follows:

Definition 13. A Σ-protocol for language L is an interactive three-move protocol between a PPT prover $\mathsf{P} = (\mathsf{Commit}, \mathsf{Prove})$ and a PPT verifier $\mathsf{V} = (\mathsf{Challenge}, \mathsf{Verify})$, where P makes the first move and transcripts are of the form $(\mathsf{a}, \mathsf{c}, \mathsf{s}) \in \mathsf{A} \times \mathsf{C} \times \mathsf{S}$. Additionally they satisfy the following properties:

Completeness: A Σ-protocol for language L is complete, if for all security parameters κ, and for all $(x, w) \in R$, it holds that

$$\Pr\left[\langle \mathsf{P}\left(1^\kappa, x, w\right), \mathsf{V}\left(1^\kappa, x\right)\rangle = 1\right] = 1.$$

s-Special Soundness: A Σ-protocol for language L is special sound, if there exists a PPT extractor $\mathcal{E}$ so that for all x, and for all sets of accepting transcripts $\{(\mathsf{a}, \mathsf{c}_i, \mathsf{s}_i)\}_{i \in [s]}$ with respect to x where $\mathsf{c}_1 \neq \mathsf{c}_2$, generated by any algorithm with polynomial runtime in κ, it holds that

$$\Pr\left[w \leftarrow \mathcal{E}\left(1^\kappa, x, \{(\mathsf{a}, \mathsf{c}_i, \mathsf{s}_i)\}_{i \in [s]}\right) \ : \ (x, w) \in R\right] \geq 1 - \varepsilon(\kappa).$$

Special Honest-Verifier Zero-Knowledge: A Σ-protocol is special honest-verifier zero-knowledge, if there exists a PPT simulator $\mathcal{S}$ so that for every $x \in L$ and every challenge $\mathsf{c} \in \mathsf{C}$, it holds that a transcript $(\mathsf{a}, \mathsf{c}, \mathsf{s})$, where $(\mathsf{a}, \mathsf{s}) \leftarrow \mathcal{S}(1^\kappa, x, \mathsf{c})$ is indistinguishable from a transcript resulting from an honest execution of the protocol.

The s-special soundness property gives an immediate bound for soundness: if no witness exists and ignoring a negligible error, then the prover can successfully answer at most to ${}^{s-1}/_{|\mathsf{C}|}$ challenges. In case this value is too large, it is possible to reduce the soundness error using ℓ-fold parallel repetition of the Σ-protocol. We recall some well known facts of Σ-protocols (cf. [Dam10; Sch19]):

Lemma 1. The properties of Σ-protocols are invariant under parallel repetition. In particular, the ℓ-fold parallel repetition of a Σ-protocol for relation R with challenge length t yields a new Σ-protocol with challenge length $\ell \cdot t$.

Lemma 2. Let L_1 and L_2 be two languages with associated witness relations R_1 and R_2, respectively. Further, let Σ_1 and Σ_2 be two Σ-protocols with identical challenge space so that Σ_1 is for L_1 and Σ_2 is for L_2. Then a Σ-protocol for the conjunction of L_1 and L_2, i.e.,

$$L_1 \wedge L_2 = \{(x_1, x_2) \mid \exists\ w_1, w_2 \colon (x_1, w_1) \in L_1 \ \wedge \ (x_2, w_2) \in L_2\}$$

is obtained by running Σ_1 and Σ_2 in parallel using a single common challenge e.

Σ-protocols can be a useful tool for different applications. Most notably, Schnorr's Σ-protocol [Sch89] allows the prover to proof knowledge of a discrete logarithm on one hand and also leads to a signature scheme on the other. The conjunction of two versions of that Σ-protocol with respect to different generators immediately yields one of for DDH-tuples, or in other words, allows us to proof statements related to ElGamal ciphertexts (cf. Scheme 3).

2.3.4. Non-Interactive Zero-Knowledge Proofs

We recall a standard definition of non-interactive zero-knowledge proof systems. As before, $L \subseteq \mathsf{X}$ be an **NP**-language with associated witness relation R.

Definition 14 (Non-Interactive Zero-Knowledge Proof System)**.** A non-interactive proof system Π is a tuple of algorithms $(\mathsf{Setup}, \mathsf{Proof}, \mathsf{Verify})$, which are defined as follows:

$\mathsf{Setup}(1^\kappa)$: This algorithm takes a security parameter κ as input, and outputs a common reference string crs.

$\mathsf{Proof}(\mathsf{crs}, x, w)$: This algorithm takes a common reference string crs, a statement x, and a witness w as input, and outputs a proof π.

$\mathsf{Verify}(\mathsf{crs}, x, \pi)$: This algorithm takes a common reference string crs, a statement x, and a proof π as input, and outputs a bit $b \in \{0, 1\}$.

From a non-interactive zero-knowledge proof system we require completeness, soundness, adaptive zero-knowledge and simulation-sound extractability.

Definition 15 (Perfect Completeness)**.** A non-interactive proof system for language L is complete, if for all $\kappa \in \mathbb{N}$, for all $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa)$, for all $x \in L$, for all w such that $R(x, w) = 1$, and for all $\pi \leftarrow \mathsf{Proof}(\mathsf{crs}, x, w)$, we have that $\mathsf{Verify}(\mathsf{crs}, x, \pi) = 1$.

Definition 16 (Soundness)**.** For an efficient adversary $\mathcal{A}$, we define the advantage function in the sense of soundness as

$$\mathsf{Adv}^{\mathsf{Sound}}_{\mathcal{A},\Pi}(\kappa) = \Pr\left[\begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa), \\ (x, \pi) \leftarrow \mathcal{A}(\mathsf{crs}) \end{array} : \begin{array}{r} \mathsf{Verify}(\mathsf{crs}, x, \pi) = 1 \\ \wedge\ x \notin L \end{array}\right].$$

If for any efficient adversary $\mathcal{A}$ there exists a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{Sound}}_{\mathcal{A},\Pi}(\kappa) \leq \varepsilon(\kappa),$$

then Π is sound.

Definition 17 (Adaptive Zero-Knowledge)**.** For an efficient simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ and an efficient adversary $\mathcal{A}$, we define the advantage functions in the sense of zero-knowledge as

$$\mathsf{Adv}^{\mathsf{Sim}}_{\mathcal{A},\mathcal{S},\Pi}(\kappa) = \left|\begin{array}{l} \Pr\left[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa) : \mathcal{A}(\mathsf{crs}) = 1\right] - \\ \Pr\left[(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^\kappa) : \mathcal{A}(\mathsf{crs}) = 1\right] \end{array}\right|$$

and

$$\mathsf{Adv}^{\mathsf{ZK}}_{\mathcal{A},\mathcal{S},\Pi}(\kappa) = \left|\Pr\left[\mathsf{Exp}^{\mathsf{ZK}}_{\mathcal{A},\mathcal{S},\Pi}(1^\kappa) = 1\right] - \frac{1}{2}\right|$$

where the corresponding experiment is depicted in Experiment 2. If there exists an efficient simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that for any efficient adversary $\mathcal{A}$ there exist negligible functions $\varepsilon_1(\cdot)$ and $\varepsilon_2(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{Sim}}_{\mathcal{A},\mathcal{S},\Pi}(\kappa) \leq \varepsilon_1(\kappa) \text{ and } \mathsf{Adv}^{\mathsf{ZK}}_{\mathcal{A},\mathcal{S},\Pi}(\kappa) \leq \varepsilon_2(\kappa)$$

then Π provides adaptive zero-knowledge.

```
$\mathsf{Exp}^{\mathsf{ZK}}_{\mathcal{A},\mathsf{S},\Pi}(1^\kappa)$:
  $b \leftarrow \{0,1\}$
  $(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^\kappa)$
  $b^* \leftarrow \mathcal{A}^{\mathsf{P}_b(\cdot,\cdot)}(\mathsf{crs})$
    where oracle $\mathsf{P}_0(x, w)$:
      return $\pi \leftarrow \mathsf{Proof}(\mathsf{crs}, x, w)$, if $(x, w) \in R$
      return $\perp$
    and oracle $\mathsf{P}_1(x, w)$:
      return $\pi \leftarrow \mathcal{S}_2(\mathsf{crs}, \tau, x)$, if $(x, w) \in R$
      return $\perp$
  return 1, if $b = b^*$
  return 0
```

Experiment 2: Adaptive zero-knowledge experiment.

Definition 18 (Simulation-Sound Extractability)**.** For an adaptively zero-knowledge non-interactive proof system Π, for an efficient extractor extractor $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$ and an efficient adversary $\mathcal{A}$, we define the advantage functions in the sense of simulation-sound extractability as

$$\mathsf{Adv}^{\mathsf{Ext}_1}_{\mathcal{A},\mathcal{E},\Pi}(\kappa) = \left| \begin{array}{l} \Pr\left[(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1\left(1^\kappa\right) : \mathcal{A}(\mathsf{crs}) = 1\right] - \\ \Pr\left[(\mathsf{crs}, \tau, \xi) \leftarrow \mathcal{E}_1\left(1^\kappa\right) : \mathcal{A}(\mathsf{crs}) = 1\right] \end{array} \right|$$

and

$$\mathsf{Adv}^{\mathsf{Ext}_2}_{\mathcal{A},\mathcal{E},\Pi}(\kappa) = \left| \Pr\left[\mathsf{Exp}^{\mathsf{Ext}_2}_{\mathcal{A},\mathcal{E},\Pi}(1^\kappa) = 1 \right] \right|,$$

where the corresponding experiment is depicted in Experiment 3. If there exists an efficient extractor $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$ such that for any efficient adversary $\mathcal{A}$ there exist negligible functions $\varepsilon_1(\cdot)$ and $\varepsilon_2(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{Ext}_1}_{\mathcal{A},\mathcal{E},\Pi}(\kappa) \leq \varepsilon_1(\kappa) \text{ and } \mathsf{Adv}^{\mathsf{Ext}_2}_{\mathcal{A},\mathcal{E},\Pi}(\kappa) \leq \varepsilon_2(\kappa)$$

then Π provides simulation-sound extractactability.

NIZK proof systems can be constructed for any **NP** language, e.g. from (uniform) one-way functions [GMW86; IY87; LFK+90; Sha90], but also from various assumptions. Most recently, Peikert et al. [PS19] presented a construction from the learning with errors assumption. Our focus is on NIZK proof systems that can be obtained from Σ-protocols. We discuss two transformations below.

NIZK from Σ-protocols. Given Σ-protocol for language L, one can obtain a non-interactive proof system with the above properties by applying the Fiat-Shamir transform [FS86] to any Σ-protocol. For that transform we require the min-entropy μ of the commitment a sent in the first message of the Σ-protocol

```
$\mathsf{Exp}^{\mathsf{Ext}_2}_{\mathcal{A},\mathcal{E},\Pi}(1^\kappa)$:
  $(\mathsf{crs}, \tau, \xi) \leftarrow \mathcal{E}_1(1^\kappa)$
  $\mathcal{Q}_\mathcal{S} \leftarrow \emptyset$
  $(x^*, w^*) \leftarrow \mathcal{A}^{\mathcal{S}(\cdot,\cdot)}(\mathsf{crs})$
    where oracle $\mathcal{S}(x, w)$:
      $\mathcal{Q}_\mathcal{S} \leftarrow \mathcal{Q}_\mathcal{S} \cup \{(x, w)\}$
      return $\pi \leftarrow \mathcal{S}_2(\mathsf{crs}, \tau, x)$, if $(x, w) \in R$
      return $\perp$
  $w \leftarrow \mathcal{E}_2(\mathsf{crs}, \xi, x^*, \pi^*)$
  return 1, if $\mathsf{Verify}(\mathsf{crs}, x^*, \pi^*) = 1 \wedge (x^*, \pi^*) \notin \mathcal{Q}_\mathcal{S} \wedge (x^*, w) \notin R$
  return 0
```

Experiment 3: Simulation-sound extractability experiment.

to be such that $2^{-\mu}$ is negligible in the security parameter κ. Furthermore, its challenge space C needs to exponentially large in the security parameter. Essentially, the transform removes the interaction between the prover and the verifier by using a hash function H (modelled as a random oracle) to obtain the challenge. That is, the algorithm $\mathsf{Challenge}$ obtains the challenge as $H(\mathsf{a}, x)$. We formally recall this stronger variant of the Fiat-Shamir transform [FKM+12; BPW12] in Scheme 1. The original variant of the transform does not include the statement x in the challenge generation.

$\mathsf{Setup}(1^\kappa)$: Choose a hash function $H : \mathsf{A} \times \mathsf{X} \rightarrow \mathsf{C}$, set $\mathsf{crs} \leftarrow (\kappa, H)$, and return crs.

$\mathsf{Proof}(\mathsf{crs}, x, w)$: Start P on $(1^\kappa, x, w)$, obtain the first message a, answer with $\mathsf{c} \leftarrow H(\mathsf{a}, x)$. Finally obtain s and return $\pi \leftarrow (\mathsf{a}, \mathsf{s})$.

$\mathsf{Verify}(\mathsf{crs}, x, \pi)$: Parse π as (a, s). Start V on $(1^\kappa, x)$ and send a as first message to the verifier. When V outputs c, reply with s and output 1 if V accepts and 0 otherwise.

Scheme 1: NIZK obtained by applying the Fiat-Shamir transform to a Σ-protocol.

Faust et al. [FKM+12] showed that a so-obtained proof system is complete, sound, adaptively zero-knowledge in the ROM, if the underlying Σ-protocol is special sound and the commitments sent in the first move are unconditionally binding. Security of the Fiat-Shamir transform in the QROM is harder to achieve. Ambainis et al. [ARU14] showed that classical results relying on rewinding do not hold in the QROM. Especially, special soundness of the Σ-protocol is no longer sufficient. We however note, that the Fiat-Shamir transform still yields secure NIZKs in the QROM if the underlying Σ-protocol satisfies stronger properties [Unr17; KLS18; DFM+19; LZ19].

To obtain a secure NIZK in the QROM we can apply less efficient transforms such as Unruh's transform [Unr15] instead. At a high level, Unruh's transform works as follows: Given a s-special-sound Σ-protocol, integers t and M, a statement x and a random permutation G, the prover will repeat the first phase of the Σ-protocol t times. Then, for each of the t runs, it produces proofs to M different randomly selected challenges. The prover applies G to each of the so-obtained responses. The prover then selects the responses to publish for each round of the Σ-protocol by querying the random oracle on the message to be signed, all first rounds of the Σ-protocol and the outputs of G on all responses.

We present NIZK obtained from Unruh's transform in Scheme 2. While the so-obtained proof system is secure in the QROM, it comes at significant overhead in runtime and in proof size by a factor of $t \cdot M$.

$\mathsf{Setup}(1^\kappa)$: Choose $t \in \mathbb{N}$, $M \in [s, |\mathsf{C}|]$, a hash function $H : \mathsf{A}^t \times \mathsf{C}^{tM} \times \mathsf{S}^{tM} \to [M]^t$, and a random permutation $G : \mathsf{S} \to \mathsf{S}$. Set $\mathsf{crs} \leftarrow (\kappa, t, M, H, G)$ and return crs.

$\mathsf{Proof}(\mathsf{crs}, x, w)$:

1. For $i \in [t]$:
   a) Start P on $(1^\kappa, x, w)$ and obtain first message a_i.
   b) For $j \in [M]$, set $\mathsf{c}_{i,j} \xleftarrow{R} \mathsf{C} \setminus \{\mathsf{c}_{i,1}, \ldots, \mathsf{c}_{i,j-1}\}$ and obtain response $\mathsf{z}_{i,j}$ for challenge $\mathsf{c}_{i,j}$.
2. For $i, j \in [t] \times [M]$, set $g_{i,j} \leftarrow G(\mathsf{z}_{i,j})$.
3. Let $(J_1, \ldots, J_t) \leftarrow H\left((\mathsf{a}_i)_{i\in[t]}, (\mathsf{c}_{i,j})_{(i,j)\in[t]\times[M]}, (g_{i,j})_{(i,j)\in[t]\times[M]}\right)$.
4. Return $\pi \leftarrow \left((\mathsf{a}_i)_{i\in[t]}, (\mathsf{c}_{i,j})_{(i,j)\in[t]\times[M]}, (g_{i,j})_{(i,j)\in[t]\times[M]}, (\mathsf{z}_{i,J_i})_{i\in[t]}\right)$.

$\mathsf{Verify}(\mathsf{crs}, x, \pi)$: Parse π as

$$\left((\mathsf{a}_i)_{i\in[t]}, (\mathsf{c}_{i,j})_{(i,j)\in[t]\times[M]}, (g_{i,j})_{(i,j)\in[t]\times[M]}, (\mathsf{z}_{i,J_i})_{i\in[t]}\right).$$

1. Let $(J_1, \ldots, J_t) \leftarrow H\left((\mathsf{a}_i)_{i\in[t]}, (\mathsf{c}_{i,j})_{(i,j)\in[t]\times[M]}, (g_{i,j})_{(i,j)\in[t]\times[M]}\right)$.
2. For $i \in [t]$ check that all $\mathsf{c}_{i,1}, \ldots, \mathsf{c}_{i,M}$ are pairwise distinct.
3. For $i \in [t]$ check whether V accepts the proof with respect to x, first message a_i, challenge c_{i,J_i} and response z_i.
4. For $i \in [t]$ check $g_{i,J_i} = G(\mathsf{z}_i)$.
5. Output 1 if all checks succeeded and 0 otherwise.

Scheme 2: NIZK obtained by applying Unruh's transform to a Σ-protocol.

2.4. Public-Key Encryption and Variants

We briefly recall syntax and $\mathsf{IND\text{-}CPA}$ security of public-key encryption.

Definition 19 (Public-Key Encryption)**.** A public-key encryption scheme Ω is a triple $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ of PPT algorithms such that:

$\mathsf{KGen}(1^\kappa)$: This algorithm on input security parameter κ outputs the secret and public key $(\mathsf{sk}, \mathsf{pk})$, where the public key pk implicitly defines the message space $\mathcal{M}$.

$\mathsf{Enc}(\mathsf{pk}, m)$: This algorithm input the public key pk, and the message $m \in \mathcal{M}$ and outputs a ciphertext c.

$\mathsf{Dec}(\mathsf{sk}, C)$: This algorithm on input a secret key sk and a ciphertext c outputs a message $m \in \mathcal{M} \cup \{\perp\}$.

We say that an encryption scheme Ω is perfectly correct if for all $\kappa \in \mathbb{N}$, for all $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\kappa)$ and for all $m \in \mathcal{M}$ it holds that

$$\Pr\left[\mathsf{Dec}\left(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)\right) = m\right] = 1.$$

Indistinguishability under chosen message attacks ($\mathsf{IND\text{-}CPA}$ security) requires that an adversary $\mathcal{A}$ cannot decide which message is actually contained in a ciphertext c even when allowed to choose two challenge messages m_0 and m_1.

Definition 20 ($\mathsf{IND\text{-}CPA}$)**.** For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of indistinguishability under chosen message attacks ($\mathsf{IND\text{-}CPA}$) as

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{A},\Omega}(1^\kappa) = \left| \Pr\left[\mathsf{Exp}^{\mathsf{ind\text{-}cpa}}_{\mathcal{A},\Omega}(1^\kappa) = 1\right] - \frac{1}{2} \right|,$$

where the corresponding experiment is depicted in Experiment 4. If for all PPT adversaries $\mathcal{A}$ there is a is a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{A},\Omega}(1^\kappa) \leq \varepsilon(\kappa),$$

then Ω is $\mathsf{IND\text{-}CPA}$ secure.

For many applications indistinguishability under chosen ciphertext attacks is a more interesting property. There, an adversary also has access to a decryption oracle which can be used to decrypt all ciphertext except the challenge ciphertext. For the following discussion of forward secrecy and proxy re-encryption, we will discuss the concepts also with respect to $\mathsf{IND\text{-}CPA}$-style security experiments, which will be enough to give an overview of the security properties.

We shortly recall ElGamal encryption [Gam84] in Scheme 3, because we build on it in both [DKL+18a; DRS18c]. This scheme is $\mathsf{IND\text{-}CPA}$-secure under the decisional Diffie-Hellman assumption. Notably, this encryption scheme is multiplicatively homomorphic. Alternatively, to obtain an additively homomorphic scheme, it is also possible to build ElGamal-style encryption from the DLIN assumption [BBS04]. The latter is also interesting when considering encryption schemes when DDH is easy, e.g. in bilinear groups.

```
$\mathsf{Exp}^{\mathsf{ind\text{-}cpa}}_{\mathcal{A},\Omega}(1^\kappa)$:
    $(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{KGen}(1^\kappa)$
    $b \xleftarrow{R} \{0,1\}$
    $(m_0, m_1, \mathtt{state}) \leftarrow \mathcal{A}(\mathsf{pk})$
    $c^* \leftarrow \mathsf{Enc}\,(\mathsf{pk}, m_b)$
    $b^* \leftarrow \mathcal{A}\,(c^*, \mathtt{state})$
    return 1, if $b^* = b$
    return 0
```

Experiment 4: IND-CPA security experiment for public-key encryption.

$\mathsf{Gen}(1^\kappa)$: Choose a group $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^\kappa)$. Choose $x \xleftarrow{R} \mathbb{Z}_q$ and set $y \leftarrow g^x$. Return public key $\mathsf{pk} \leftarrow (\mathsf{pp}, y)$ and secret key $\mathsf{sk} \leftarrow (\mathsf{pp}, x)$.

$\mathsf{Enc}(\mathsf{pk}, m)$: Parse the public key pk as (pp, y), and message $m \in \mathbb{G}$, choose $r \xleftarrow{R} \mathbb{Z}_q$ and output $c \leftarrow (g^r, M \cdot y^r)$.

$\mathsf{Dec}(\mathsf{sk}, c)$: Parse the secret key sk as (pp, x), the ciphertext c as (c_1, c_2), and output $m \leftarrow c_2 \cdot c_1^{-x}$.

Scheme 3: ElGamal public-key encryption.

2.4.1. Forward-Secret Public-Key Encryption

Forward-secret public-key encryption is an extension of classical public-key encryption which attaches a time period to secret keys and ciphertexts. It also adds an additional algorithm to evolve secret keys from one period to the next. Yet, the public key stays constant throughout its lifetime, and additionally its size is sublinear in the number of periods. The goal is that even if an adversary is in possession of a secret for an period, it is impossible to learn anything about ciphertexts of previous periods.

Definition 21 (Forward-Secret Public-Key Encryption)**.** A forward-secret public-key encryption scheme fs-Ω is a tuple $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Update})$ of PPT algorithms such that:

$\mathsf{KGen}(1^\kappa, n)$: This algorithm on input security parameter κ, and a maximal number of periods n, and outputs the secret and public key $(\mathsf{sk}, \mathsf{pk})$.

$\mathsf{Enc}(\mathsf{pk}, m, j)$: This algorithm input the public key pk, the message $m \in \mathcal{M}$, and an period $j \in [n]$ and outputs a ciphertext c.

$\mathsf{Dec}(\mathsf{sk}, C)$: This algorithm on input a secret key sk and a ciphertext c, both for the same period $j \in [n]$, outputs a message $m \in \mathcal{M} \cup \{\perp\}$.

$\mathsf{Update}(\mathsf{sk})$: This algorithm in input a secret key sk for period $j \in [n-1]$, outputs a secret key sk' for period $j + 1$.

For correctness, we require that for all $\kappa \in \mathbb{N}$, for all $n \leq \mathsf{poly}(\kappa)$, for all $(\mathsf{sk}^{(0)}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\kappa, n)$, for all $j \in [n]$, and for all $m \in \mathcal{M}$ it holds that

$$\Pr\left[\mathsf{Dec}\left(\mathsf{sk}^{(j)}, \mathsf{Enc}(\mathsf{pk}, m, j)\right) = m\right] = 1,$$

where $\mathsf{sk}^{(j)}$ is obtained by computing $\mathsf{sk}^{(i+1)} \leftarrow \mathsf{Update}(\mathsf{sk}^{(i)})$ for $i \in [j-1]$.

To adapt the IND-CPA notion, first of all the adversary has to select a target period. The challenge ciphertext is then encrypted for the penultimate period and the adversary gets access to the secret key of the target period.

Definition 22 (fs-IND-CPA)**.** For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of forward-secret indistinguishability under chosen message attacks (fs-IND-CPA) as

$$\mathsf{Adv}^{\text{fs-ind-cpa}}_{\mathcal{A},fs-\Omega}(1^\kappa, n) = \left|\Pr\left[\mathsf{Exp}^{\text{fs-ind-cpa}}_{\mathcal{A},fs-\Omega}(1^\kappa, n) = 1\right] - \frac{1}{2}\right|,$$

where the corresponding experiment is depicted in Experiment 5. If for all PPT adversaries $\mathcal{A}$ and $n \in [\mathsf{poly}(\kappa)]$, there is a is a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\text{fs-ind-cpa}}_{\mathcal{A},fs-\Omega}(1^\kappa) \leq \varepsilon(\kappa),$$

then Ω is fs-IND-CPA secure.

```
$\mathsf{Exp}^{\text{fs-ind-cpa}}_{\mathcal{A},fs-\Omega}(1^\kappa, n)$:
    $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\kappa, n)$
    $b \xleftarrow{R} \{0, 1\}$
    $(m_0, m_1, j^*, \mathtt{state}) \leftarrow \mathcal{A}(\mathsf{pk})$
    $\mathsf{sk}^{(i+1)} \leftarrow \mathsf{Update}\left(\mathsf{sk}^{(i)}\right)$ for $i \in [j-1]$
    $c^* \leftarrow \mathsf{Enc}\left(\mathsf{pk}, m_b, j^* - 1\right)$
    $b^* \leftarrow \mathcal{A}\left(c^*, \mathsf{sk}^{(j^*)}, \mathtt{state}\right)$
    return 1, if $b^* = b$
    return 0
```

Experiment 5: fs-IND-CPA security experiment for forward-secret public-key encryption.

We shortly discuss a generic tree-based approach.

The Canetti-Halevi-Katz compiler. To build a forward-secret scheme with n periods, Canetti et al. [CHK03] attach periods to the nodes of a depth ℓ binary tree with $n < 2^\ell$. The periods are arranged in depth-first manner in the tree and the key update algorithm. Now, the public key is simply the public key of a binary-tree or hierarchical identity-based encryption scheme. Both schemes allow one to derive secret keys arranged in a tree such that keys can only be derived from the root downwards, but not in the other direction. The secret key for a period then consists of the secret key of the associated node and the secret keys for all nodes to be able to traverse the tree in a depth-first manner.

The more abstract concept of puncturable encryption [GM15; GHJ+17] allows secret keys to be evolved in a way, that those keys can no longer be used to decrypt certain ciphertexts, e.g. ciphertexts that have been encrypted with respect to a tag. When mapping time periods to tags, puncturable encryption schemes trivially imply forward-secret public-key encryption. However, the currently known constructions either explicitly or implicitly use the techniques inspired by Canetti, Halevi, and Katz.

2.4.2. Proxy Re-Encryption

Proxy re-encryption, envisioned by Blaze et al. [BBS98] and formalized by Ateniese, Fu, Green, and Hohenberger [AFG+05; AFG+06], can be seen as an extension of public-key encryption. A central feature of proxy re-encryption is that senders can craft so-called re-encryption keys, which are usually created using only public information of the designated delegatee and the delegators' key material. Those re-encryption keys have the power to transform ciphertexts under a delegator's public key to ciphertexts under the delegatees' public keys. Within proxy re-encryption, this transformation is done by a semi-trusted proxy. The widely accepted model for secure proxy re-encryptions [AFG+05] requires that the proxy does not learn anything about the plaintexts which are encrypted in the ciphertexts intended to be transformed. Proxy re-encryption is considered very useful in applications such as encrypted e-mail forwarding or access control in secure file systems, which was already discussed in earlier work, e.g., in [AFG+05].

We recall the standard definition of proxy re-encryption [AFG+05; AFG+06; LV08b] focusing on the uni-directional, single-hop variant.

Definition 23 (Proxy Re-Encryption)**.** A proxy re-encryption (PRE) scheme with message space $\mathcal{M}$ consists of the PPT algorithms ($\mathsf{Setup}, \mathsf{Gen}, \vec{\mathsf{Enc}}, \vec{\mathsf{Dec}}, \mathsf{ReGen}, \mathsf{ReEnc}$) where $\vec{\mathsf{Enc}} = (\mathsf{Enc}^{(j)})_{j\in[2]}$ and $\vec{\mathsf{Dec}} = (\mathsf{Dec}^{(j)})_{j\in[2]}$. For $j \in [2]$, they are defined as follows.

$\mathsf{Setup}(1^\kappa)$: On input security parameter κ, outputs public parameters pp.

$\mathsf{Gen}(\mathsf{pp})$: On input public parameters pp, outputs public and secret keys $(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{Enc}^{(j)}(\mathsf{pk}, m)$: On input a public key pk, and a message $m \in \mathcal{M}$ outputs a level j ciphertext c.

$\mathsf{Dec}^{(j)}(\mathsf{sk}, c)$: On input a secret key sk, and level j ciphertext c, outputs $m \in \mathcal{M} \cup \{\perp\}$.

$\mathsf{ReGen}(\mathsf{sk}_A, \mathsf{pk}_B)$: On input a secret key sk_A and a public key pk_B for B, outputs a re-encryption $\mathsf{rk}_{A\to B}$.

$\mathsf{ReEnc}(\mathsf{rk}_{A\to B}, c_A)$: On input a re-encryption key $\mathsf{rk}_{A\to B}$, and a ciphertext c_A for user A, outputs a ciphertext c_B for user B.

For correctness we require that for all security parameters $\kappa \in \mathbb{N}$, all public parameters $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa)$, any number of users $U \in \mathbb{N}$, all key tuples $(\mathsf{pk}_u, \mathsf{sk}_u)_{u\in[U]}$ generated by $\mathsf{Gen}(1^\kappa)$, for any $u, u' \in [U], u \neq u'$, any re-encryption keys $\mathsf{rk}_{u\to u'}$ using ReGen, and all messages $m \in \mathcal{M}$, it holds that

$$\forall j \in [2]\, \exists j' \in [2]\colon \Pr\left[\mathsf{Dec}^{(j')}\left(\mathsf{sk}_u, \mathsf{Enc}^{(j)}(\mathsf{pk}_u, m)\right) = m\right] = 1, \text{ and}$$
$$\Pr\left[\mathsf{Dec}^{(1)}\left(\mathsf{sk}_{u'}, \mathsf{ReEnc}\left(\mathsf{rk}_{u\to u'}, \mathsf{Enc}^{(2)}(\mathsf{pk}_u, m)\right)\right) = m\right] = 1.$$

We stress that level-2 ciphertexts are re-encryptable ciphertexts, whereas level-1 ciphertexts are not re-encryptable and we only consider single-hop proxy re-encryption.

Subsequently we discuss IND-CPA style security experiments for level 1 and level 2 ciphertexts. Both experiments share oracles, so we discuss them here. For all experiments defined in this section, the environment keeps initially empty lists of dishonest (DU) and honest users (HU).

$\mathsf{Gen}^{(h)}(\mathsf{pp})$: Run $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{pp})$, set $\mathtt{HU} \leftarrow \mathtt{HU} \cup \{(\mathsf{pk}, \mathsf{sk})\}$, and return pk.

$\mathsf{Gen}^{(d)}(\mathsf{pp})$: Run $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{pp})$, set $\mathtt{DU} \leftarrow \mathtt{DU} \cup \{(\mathsf{pk}, \mathsf{sk})\}$, and return $(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{ReGen}^{(h)}(\mathsf{pk}_u, \mathsf{pk})$: On input a public key pk_u and a public key pk, abort if $(\mathsf{pk}_u, \cdot) \notin \mathtt{HU}$. Otherwise, look up sk_u corresponding to pk_u from $\mathtt{HU}$. Return $\mathsf{ReGen}(\mathsf{sk}_u, \mathsf{pk})$.

$\mathsf{ReGen}^{(h')}(\mathsf{sk}, \mathsf{pk}_u)$: On input a secret key sk and a public key pk_u, abort if $(\mathsf{pk}_u, \cdot) \notin \mathtt{HU}$. Otherwise, return $\mathsf{ReGen}(\mathsf{sk}, \mathsf{pk}_u)$.

$\mathsf{ReGen}^{(d)}(\mathsf{sk}, \mathsf{pk}_d)$: On input a secret key sk and a public key pk_d, abort if $(\mathsf{pk}_d, \cdot) \notin \mathtt{DU}$. Otherwise, return $\mathsf{ReGen}(\mathsf{sk}, \mathsf{pk}_d)$.

We now recall the definition of IND-CPA security for level 1 ciphertexts. Since level 1 ciphertexts are not re-encryptable, re-encryption keys should be of no use to break indistinguishability. Hence, the adversary can request all possible re-encryption keys in this case.

Definition 24 (IND-CPA-1)**.** For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of IND-CPA for level 1 ciphertexts as

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa\text{-}1}}_{\mathcal{A},\mathsf{PRE}}(1^\kappa) = \left|\Pr\left[\mathsf{Exp}^{\mathsf{ind\text{-}cpa\text{-}1}}_{\mathcal{A},\mathsf{PRE}}(1^\kappa) = 1\right] - \frac{1}{2}\right|,$$

where the corresponding experiment is depicted in Experiment 6. If for all PPT adversary $\mathcal{A}$, there exists a negligible function ε such that

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa\text{-}1}}_{\mathcal{A},\mathsf{PRE}}(1^\kappa) \leq \varepsilon(\kappa),$$

then a PRE scheme is IND-CPA-1 secure.

```
$\mathsf{Exp}^{\mathsf{ind\text{-}cpa\text{-}1}}_{\mathcal{A},\mathsf{PRE}}(1^\kappa)$:
    $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa)$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{pp})$
    $b \xleftarrow{R} \{0, 1\}$
    $\mathcal{O} \leftarrow \{\mathsf{Gen}^h, \mathsf{ReGen}^h(\cdot, \mathsf{pk}), \mathsf{ReGen}^{h'}(\mathsf{sk}, \cdot), \mathsf{Gen}^d, \mathsf{ReGen}^d(\mathsf{sk}, \cdot)\}$
    $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pk})$
    $c^* \leftarrow \mathsf{Enc}^{(1)}(\mathsf{pk}, m_b)$
    $b^* \leftarrow \mathcal{A}(\mathsf{st}, c^*)$
    return 1 if $b = b^*$
    return 0
```

Experiment 6: IND-CPA security experiment for level 1 ciphertexts for PRE.

We also recall the definition of IND-CPA security of level 2 ciphertexts. Note that here queries to the ReGen oracles are more restricted to avoid queries of re-encryption keys that would trivially break indistinguishability.

Definition 25 (IND-CPA-2)**.** For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of IND-CPA for level 2 ciphertexts as

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa\text{-}2}}_{\mathcal{A},\mathsf{PRE}}(1^\kappa) = \left| \Pr\left[\mathsf{Exp}^{\mathsf{ind\text{-}cpa\text{-}2}}_{\mathcal{A},\mathsf{PRE}}(1^\kappa) = 1\right] - \frac{1}{2} \right|,$$

where the corresponding experiment is depicted in Experiment 7. If for all PPT adversary $\mathcal{A}$, there exists a negligible function ε such that

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa\text{-}2}}_{\mathcal{A},\mathsf{PRE}}(1^\kappa) \leq \varepsilon(\kappa),$$

then a PRE scheme is IND-CPA-2 secure.

Besides proxy re-encryption in the classical setting, it has been object of significant research for almost two decades, including proxy re-encryption with temporary delegation [AFG+05; AFG+06; LV11], identity-based proxy re-encryption [GA07; RGW+10], extensions to the chosen-ciphertext setting [CH07; LV11], type-based/conditional proxy re-encryption [Tan08; WYT+09], anonymous (or key-private) proxy re-encryption [ABH09], traceable proxy re-encryption [LV08a], or proxy re-encryption from lattice-based assumptions [CCL+14; PRS+17].

We will discuss the basic ideas of the constructions of Ateniese et al. [AFG+05].

Shifting keys. Note that in the bilinear group settings one can produce ElGamal-style ciphertexts both in the source groups and in the target group. Additionally, by applying the pairing, ciphertexts in the source group can be transformed into ciphertexts in the target group. So we immediately have obvious choices for re-encryptable ciphertexts in the source groups and non-re-

```
$\mathsf{Exp}^{\mathsf{ind\text{-}cpa\text{-}2}}_{\mathcal{A},\mathsf{PRE}}(1^\kappa)$:
    $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa)$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{pp})$
    $b \xleftarrow{R} \{0, 1\}$
    $\mathcal{O} \leftarrow \{\mathsf{Gen}^h, \mathsf{ReGen}^h(\cdot, \mathsf{pk}), \mathsf{ReGen}^{h'}(\mathsf{sk}, \cdot)\}$
    $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pk})$
    $c^* \leftarrow \mathsf{Enc}^{(2)}(\mathsf{pk}, m_b)$
    $b^* \leftarrow \mathcal{A}(\mathsf{st}, c^*)$
    return 1 if $b = b^*$
    return 0
```

Experiment 7: IND-CPA security experiment for level 2 ciphertexts for PRE.

encryptable ciphertexts in the target group, respectively. Moreover, given that the pairings enables us to perform one multiplication in the exponent. Therefore, the construction of Ateniese et al. [AFG+05] computes re-encryption keys as public key of the receiver raised to the power of the inverse of the secret key of the sender. When then pairing the re-encryption key with the ciphertext components, the public key of the sender gets canceled out.

A weak form of forward-secret proxy re-encryption can be achieved using temporary delegations as proposed by Ateniese et al. [AFG+05; AFG+06] and later improved by Libert et al. [LV11]. Alternatively, it can also be achieved by using type-based/conditional proxy re-encryption [Tan08; WYT+09]. However, both approaches at least require to update the re-encryption keys for each time period with the help of the delegator. To get a stronger form of forward-secret proxy re-encryption, we first adapt the notion of forward-secrecy to the setting of proxy re-encryption. In

- David Derler, Stephan Krenn, Thomas Lorünser, Sebastian Ramacher, Daniel Slamanig, and Christoph Striecks. Revisiting Proxy Re-encryption: Forward Secrecy, Improved Security, and Applications. In Michel Abdalla and Ricardo Dahab, editors, *Public-Key Cryptography - PKC 2018 - 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part I*, volume 10769 of *Lecture Notes in Computer Science*, pages 219–250. Springer, 2018. URL: https://doi.org/10.1007/978-3-319-76578-5_8,

we model forward-secrecy both on the sender's and the proxy's side. We then continue by proposing a construction of forward-secret proxy re-encryption. For this scheme we employ the CHK compiler [CHK03] combined with homomorphic encryption in a way that allows us to evolve encrypted secret keys in the encrypted domain.

As an intermediate step, we introduce the notion of a forward-secret delegatable public-key encryption scheme. Such a scheme allows the delegatee to delegate the decryption functionality to some other user by computing a delegation key. This key essentially contains an encrypted version of the delegatee's secret key encrypted for the receiver. To obtain forward secrecy, we combine the results of CHK with a suitable homomorphic public-key encryption scheme, so that we can perform the key derivation in the encrypted domain. We then observe that forward-secret delegatable public-key encryption built from binary-tree encryption provides key-homomorphisms and the ability to adopt ciphertexts to a shifted key. Thus, we adapt the key-shifting technique of Ateniese et al., but instead of shifting ciphertexts from one user to another, we shift the ciphertext to a newly sampled public key. Additionally, we also shift the delegation key, which serves as re-encryption key, to match the newly created ciphertext. By shifting to the newly sampled key, the connection of the encrypted key and the original secret key is gets broken up, thus ensuring that the encrypted keys leak no information on the original keys.

2.5. Digital Signatures and Variants

We recall the notion of digital signature schemes and the standard unforgeability notions below.

Definition 26 (Signature Scheme)**.** A signature scheme Σ is a triple (KGen, $\mathsf{Sign}, \mathsf{Verify}$) of PPT algorithms, which are defined as follows:

$\mathsf{KGen}(1^\kappa)$: This algorithm takes a security parameter κ as input and outputs a secret (signing) key sk and a public (verification) key pk with associated message space $\mathcal{M}$ (we may omit to make the message space $\mathcal{M}$ explicit).

$\mathsf{Sign}(\mathsf{sk}, m)$: This algorithm takes a secret key sk and a message $m \in \mathcal{M}$ as input and outputs a signature σ.

$\mathsf{Verify}(\mathsf{pk}, m, \sigma)$: This algorithm takes a public key pk, a message $m \in \mathcal{M}$ and a signature σ as input and outputs a bit $b \in \{0, 1\}$.

We require a signature scheme to be correct and to provide existential unforgeability under adaptively chosen message attacks ($\mathsf{EUF\text{-}CMA}$ security). For correctness we require that for all $\kappa \in \mathbb{N}$, for all $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\kappa)$ and for all $m \in \mathcal{M}$ it holds that

$$\Pr\left[\mathsf{Verify}(\mathsf{pk}, m, \mathsf{Sign}(\mathsf{sk}, m)) = 1\right] = 1.$$

Definition 27 ($\mathsf{EUF\text{-}CMA}$)**.** For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of existential unforgeability under chosen message attacks ($\mathsf{EUF\text{-}CMA}$) as

$$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\Sigma}(1^\kappa) = \Pr\left[\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\Sigma}(1^\kappa) = 1\right],$$

where the corresponding experiment is depicted in Experiment 8. If for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\Sigma}(1^\kappa) \leq \varepsilon(\kappa),$$

we say that Σ is $\mathsf{EUF\text{-}CMA}$ secure.

```
$\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\Sigma}(1^\kappa)$:
  $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\kappa)$
  $\mathcal{Q} \leftarrow \emptyset$
  $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}'(\mathsf{sk},\cdot)}(\mathsf{pk})$
    where oracle $\mathsf{Sign}'(m)$:
      $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$
      $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
      return $\sigma$
  return 1, if $\mathsf{Verify}(\mathsf{pk}, m^*, \sigma^*) = 1 \ \wedge \ m^* \notin \mathcal{Q}$
  return 0
```

Experiment 8: $\mathsf{EUF\text{-}CMA}$ security experiment for Σ.

In the following we will focus on two common construction techniques popular in the ROM: signatures from identification schemes and the full-domain-hash (FDH) paradigm. Both approaches have been used both in the classical and the post-quantum setting and are also used as part of the currently standardized signature schemes.

Identification schemes. Identification schemes are interactive protocols that allows a party to prove its identity to another party. These schemes can have multiple rounds whereas the three round identification schemes share a similarity with Σ-protocols. Indeed, using a suitable hard relation, three-move identification schemes can be obtained from Σ-protocols. Especially three-move identification schemes can be transformed into an $\mathsf{EUF\text{-}CMA}$ secure signature scheme by applying the Fiat-Shamir transform and including the message in the challenge generation. This technique has been thoroughly studied in the last decades, e.g. in [OO98; PS96; AAB$^+$02; AFL$^+$12; KMP16; BPS16; DGV$^+$16].

Constructions based on this paradigm are numerous, and we only mention some notable instances for further reading here. The first constructions following this technique dates back to the seminal work of Schnorr [Sch89].[4] For code-based signatures, the most prominent examples are the identification schemes due to Stern [Ste93] and Véron [Vér96]. Chen et al. [CHR$^+$16] proposed a post-quantum signature scheme whose security is based on the problem of solving a multivariate system of quadratic equations. Their scheme is obtained by building upon the 5-pass (or 3-pass) identification scheme in [SSH11] and applying the

[4] (EC)DSA [KSD13] and EdDSA [BDL$^+$12] essentially follow a similar design strategy. However, to avoid conflicts with patents granted for Schnorr's protocol, the constructions are slightly modified. In particular for (EC)DSA, this fact makes the provable security analysis more complicated. Only recently Fersch et al. [FKP16] proved ECDSA secure in the bijective ROM. Previous proofs of security covered modified variants [MS02] or proofs in the generic group model [Bro05].

Fiat-Shamir transform. From an identification scheme based on the supersingular isogeny problem [FJP14], Yoo et al. [YAJ+17] and Galbraith et al. [GPS17] proposed post-quantum signature schemes. Finally, we mention two recent examples of lattice-based post-quantum signature schemes submitted to the NIST PQC project, qTESLA [ABB+19] and Dilithium [DKL+18b], that follow a variant of this approach using Fiat-Shamir with aborts [Lyu09].

Full domain hash. Signature schemes from trapdoor one-way functions can be constructed elegantly from the full-domain-hash paradigm [BR93]. There, the public key consists of the trapdoor whereas the secret key consists of the inverse, i.e. the trapdoor. For signing, the message is first hashed and the inverse of the hash image with respect to the one way function is computed using the trapdoor. The result of this operation is then published. For verification, the one-way function is evaluated on the signature and checked against the hash of the message. The most prominent example of FDH-style signatures include those built from the RSA trapdoor permutation, i.e. RSA-FDH and RSA-PSS [BR96], whereas the latter uses a slightly tweaked approach and is standardized as part of PKCS #1 v2.1 [JK03]. Also, the BLS signature scheme [BLS01] can be viewed as application of this paradigm. For BLS signatures, bilinear pairings are required to perform the verification, i.e. to check the image of the signature under the one-way function against the hash of the message. We note that the FDH paradigm is also secure in the QROM [Zha12], so given a suitable trapdoor permutation secure against a quantum adversary implies a secure signature scheme in the QROM.

When considering signatures solely from symmetric-key primitives, the FDH paradigm does not apply due to the lack of a suitable trapdoor permutation. Example of signature schemes solely relying on the security from symmetric-key primitives, i.e. without relying on structured hardness assumptions, include hash-based schemes such as Lamport's [Lam79] or Winternitz' [DSS05] one-time signature schemes. They can then be lifted to signature schemes using Merkle trees [Mer89]. Highly efficient schemes like XMSS [BDH11] are stateful, which seems to be problematic for practical applications [MKF+16]. Stateless schemes like SPHINCS [BHH+15] avoid this issue at the cost of reduced efficient and increased signature sizes. None of these signature schemes follows one of the outlined approaches above, yet given a one-way function built from symmetric-key primitives and a suitable Σ-protocol instantiating one would directly obtain a signature scheme. However, such signature would not be efficient. Starting with ZKBoo [GMO16], Σ-protocols emerged enabling efficient proofs of arithmetic circuits. In

- Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *Proceedings of the 2017 ACM SIGSAC Confer-*

ence on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017, pages 1825–1842. ACM, 2017. URL: https://doi.org/10.1145/3133956.3133997

we improve ZKBoo and introduce ZKB++, with proof sizes reduced to roughly a half. One of the major factor contributing to the size of so-obtained signatures is the choice of the one-way function. Thus we study multiple choices and settle on LowMC [ARS+15; ARS+16] to minimize the multiplicative complexity of the circuits, and thus the signature size. We also provide an implementation to show the practical efficiency of our signature scheme. To obtain a scheme also secure in the QROM, we investigate and optimize the Unruh transform when applied to ZKB++.[5] Our optimizations heavily rely on the fact that responses share parts of the views with the other possible responses, hence the responses can be interleaved.

2.5.1. Ring Signatures

Ring signatures [RST01] are a variant a digital signatures, which allow a member of an ad-hoc group $\mathcal{R}$ defined by the member's public keys, to anonymously sign a message on behalf of $\mathcal{R}$. Such a signature attests that a member of $\mathcal{R}$ produced the signature, but the actual signer remains anonymous with respect to $\mathcal{R}$. This anonymity feature turn ring signatures into an interesting tool for various applications, such as whistleblowing as envisioned by Rivest et al.

We formally define ring signature schemes following Bender et al. [BKM09].

Definition 28 (Ring Signature)**.** A ring signature scheme RS is a tuple $(\mathsf{Setup}, \mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ of PPT algorithms, which are defined as follows:

$\mathsf{Setup}(1^\kappa)$: This algorithm takes as input a security parameter κ and outputs public parameters pp.

$\mathsf{KGen}(\mathsf{pp})$: This algorithm takes as input parameters pp and outputs a keypair $(\mathsf{sk}, \mathsf{pk})$.

$\mathsf{Sign}(\mathsf{sk}_i, m, \mathcal{R})$: This algorithm takes as input a secret key sk_i, a message $m \in \mathcal{M}$ and a ring $\mathcal{R} = (\mathsf{pk}_j)_{j \in [n]}$ of n public keys such that $\mathsf{pk}_i \in \mathcal{R}$. It outputs a signature σ.

$\mathsf{Verify}(m, \sigma, \mathcal{R})$: This algorithm takes as input a message $m \in \mathcal{M}$, a signature σ and a ring $\mathcal{R}$. It outputs a bit $b \in \{0, 1\}$.

A secure ring signature scheme needs to be correct, unforgeable, and anonymous. While we omit the obvious correctness definition, we provide formal definitions for the remaining properties. We note that Bender et al. [BKM09] have formalized multiple variants of these properties, where we always use the strongest one.

[5] Recent work by Don et al. [DFM+19] suggests that at least for ZKBoo the Fiat-Shamir transform is sufficient to obtain security in the QROM.

For unforgeability we require that without knowing any secret key sk_i corresponding to one of the public key $\mathsf{pk}_i \in \mathcal{R}$, it is infeasible to produce valid signatures with respect to arbitrary rings $\mathcal{R}$. Below we recall unforgeability with respect to insider corruption, which is the strongest unforgeability notion defined in [BKM09].

Definition 29 (Unforgeability)**.** For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of unforgeability as

$$\mathsf{Adv}^{\mathsf{unf}}_{\mathcal{A},\mathsf{RS}}(1^\kappa, n) = \Pr\left[\mathsf{Exp}^{\mathsf{unf}}_{\mathcal{A},\mathsf{RS}}(1^\kappa, n) = 1\right],$$

where the corresponding experiment is depicted in Experiment 9. If for all PPT adversaries $\mathcal{A}$ and all $n \leq \mathsf{poly}(\kappa)$, there is a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\mathsf{RS}}(1^\kappa, n) \leq \varepsilon(\kappa),$$

we say that RS is unforgeable.

```
$\mathsf{Exp}^{\mathsf{unf}}_{\mathcal{A},\mathsf{RS}}(1^\kappa, n)$:
  $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa)$
  $(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{KGen}(\mathsf{pp})$ for $i \in [n]$
  $\mathcal{Q}_\mathcal{S} \leftarrow \emptyset$
  $\mathcal{Q}_\mathcal{K} \leftarrow \emptyset$
  $(m^*, \sigma^*, \mathcal{R}^*) \leftarrow \mathcal{A}^{\mathsf{Sign}'(\cdot,\cdot,\cdot),\mathsf{KGen}'(\cdot)}((\mathsf{pk}_i)_{i\in[n]})$
    where oracle $\mathsf{Sign}'(i, m, \mathcal{R})$:
      return $\perp$ if $i \notin [n] \ \vee\ \mathsf{pk}_i \notin \mathcal{R}$
      $\mathcal{Q}_\mathcal{S} \leftarrow \mathcal{Q}_\mathcal{S} \cup \{m\}$
      return $\mathsf{Sign}(\mathsf{sk}_i, m, \mathcal{R})$
    and where oracle $\mathsf{KGen}'(i)$:
      $\mathcal{Q}_\mathcal{K} \leftarrow \mathcal{Q}_\mathcal{K} \cup \{i\}$
      return $\mathsf{sk}_i$
  return 1, if $\mathsf{Verify}(\mathsf{pk}, m^*, \sigma^*) = 1 \ \wedge\ m^* \notin \mathcal{Q}_\mathcal{S} \ \wedge\ \mathcal{R}^* \subset \{\mathsf{pk}_i\}_{i\in[n]\setminus\mathcal{Q}_\mathcal{K}}$
  return 0
```

Experiment 9: Unforgeability for RS.

Anonymity requires that it is infeasible to tell which ring member produced a certain signature as long as there are at least two honest members in the ring. We recall the strongest notion defined in [BKM09]: anonymity against full key exposure.

Definition 30 (Anonymity)**.** For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of anonymity as

$$\mathsf{Adv}^{\mathsf{anon}}_{\mathcal{A},\mathsf{RS}}(1^\kappa, n) = \left|\Pr\left[\mathsf{Exp}^{\mathsf{anon}}_{\mathcal{A},\mathsf{RS}}(1^\kappa, n) = 1\right] - \frac{1}{2}\right|,$$

where the corresponding experiment is depicted in Experiment 10. If for all PPT adversaries $\mathcal{A}$ and all $n \leq \mathsf{poly}(\kappa)$, there is a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\mathsf{RS}}(1^\kappa, n) \leq \varepsilon(\kappa),$$

we say that RS provides anonymity.

```
$\mathsf{Exp}^{\mathsf{anon}}_{\mathcal{A},\mathsf{RS}}(1^\kappa, n)$:
  $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa)$
  $(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{KGen}(\mathsf{pp})$ for $i \in [n]$
  $b \xleftarrow{R} \{0,1\}$
  $(m, j_0, j_1, \mathcal{R}, \mathsf{st}) \leftarrow \mathcal{A}^{\mathsf{Sign}'(\cdot,\cdot,\cdot)}((\mathsf{pk}_i)_{i\in[n]})$
    where oracle $\mathsf{Sign}'(i, m, \mathcal{R})$:
      return $\mathsf{Sign}(\mathsf{sk}_i, m, \mathcal{R})$
  $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}_{j_b}, m, \mathcal{R})$
  $b^* \leftarrow \mathcal{A}^{\mathsf{Sign}'(\cdot,\cdot,\cdot)}(\mathsf{st}, \sigma, (\mathsf{sk}_i)_{i\in[n]})$
  return 1 if $b = b^* \ \wedge \ \{\mathsf{pk}_{j_0}, \mathsf{pk}_{j_1}\} \subset \mathcal{R}$
  return 0
```

Experiment 10: Anonymity experiment for RS.

The two main lines of more recent work in the design of ring signatures target reducing the signature size or removing the requirement for random oracles. We discuss some of generic frameworks resulting from those works below.

Ring trapdoor functions. Brakerski and Kalai [BK10] introduce a framework for constructing $\mathsf{EUF\text{-}CMA}$ signatures from weaker unforgeability notions and observe that the transformation also holds for ring signatures. Their reduction is built on top of chameleon hash functions [KR00], which are collision resistant hash functions with a trapdoor to efficiently sample a collision. For the weaker ring signature construction so-called ring trapdoor functions, a generalization of trapdoor functions, are introduced. There, given functions $(f_i)_{i\in[n]}$ and y it is hard to find $(x_i)_{i\in[n]}$ such that $\sum_{i=1}^{n} f_i(x_i) = y$. However, when having a trapdoor for any of the f_i is easy to find $(x_i)_{i\in[n]}$ for all f_i. Furthermore, given pre-images it is not possible to tell for which f_i a trapdoor was used. A ring signature than essentially contains the pre-image for a y in the public key computed from a trapdoor for one f_i and randomly selecting all others. Ring trapdoor functions can be instantiated from the computational bilinear Diffie-Hellman assumption and from the SIS assumption.

Encrypted signatures. In Bender et al.'s construction [BKM09] all users poses key pairs of a public-key encryption scheme and a standard signature scheme. On a high level, ring signatures are then generated by first producing a signature with the user's signing key and then encrypting this signature blinded with respect to all encryption public keys. The signature then contains the encrypted signature together with a proof of knowledge of a signature that verifies under the user's public key and knowledge of the randomness for encrypting the signature. This proof is performed over the disjunction of the statement repeated for every verification key. This construction can for example be instantiated with Waters [Wat05] or Camenisch-Lysyanskaya [CL04] signatures.

Key-homomorphic signatures. Derler and Slamanig [DS16] introduce a generic construction that allows one to construct ring signatures from any key-homomorphic EUF-CMA secure signature scheme with adaptable signatures. This type of signature scheme provides an homomorphism between the secret and public key space and it additionally also allow one to shift signatures generated with respect to a public key by a delta given in the secret key space. Adaptability ensures that freshly generated and shifted signatures are indistinguishable. A ring signatures consists of a signature of the underlying scheme generated by a random key and a disjunctive proof of knowledge of the shift amount of the public keys contained in the ring and the randomly sampled key. The class of suitable signature schemes includes Schnorr [Sch89], Guillou-Qisquater [GQ88], BLS [BLS01], Katz-Wang [KW03], Waters [Wat05], PS [PS16] and randomizable structure preserving signatures [AGO$^+$14; Gha16].

Accumulators. Dodis et al. [DKN$^+$04] use an accumulator with one-way domain to accumulate the set of public keys which are the image of a one-way function under the respective secret key. They combine a proof of knowledge of a witness of one public key and knowledge of the pre-image of the corresponding secret key to obtain in a ring signature scheme. Dodis et al. present an instantiation of a strong RSA assumption-based accumulator whereas Libert et al. [LLN$^+$16] provide a construction under lattice assumptions.

When only relying on symmetric-key primitives, the only suitable choice to construct ring signatures based on those generic constructions is the accumulator-based approach. So, at first we have to select an accumulator with one-way domain. In this setting, Merkle trees [Mer89] combined with an one-way function leads to an accumulator with one-way domain that is equipped with logarithmic size membership proofs. However, to break the anonymity of the ring signatures schemes it would be enough to know the authentication path. While a disjunctive proof of knowledge over all accumulated elements would hide the path taken through the tree, this relation comes at the cost of at least linearly sized proofs. One of the main technical tools used by Dodis et al. to obtain zero-knowledge membership proofs of constant size is to exploit a property of the accumulator which is called quasi-commutativity. Yet, such a property requires some underlying algebraic structure, which we want to avoid to remain in the symmetric setting. Consequently, we design a relation in

- David Derler, Sebastian Ramacher, and Daniel Slamanig. Post-Quantum

> Zero-Knowledge Proofs for Accumulators with Applications to Ring Signatures from Symmetric-Key Primitives. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings*, volume 10786 of *Lecture Notes in Computer Science*, pages 419–440. Springer, 2018. URL: https://doi.org/10.1007/978-3-319-79063-3_20

that allows us to emulate quasi-commutativity and hence to obtain zero-knowledge membership proofs of logarithmic size. To achieve that, we perform disjunctive proofs for each level of the tree and exploit the disjunction to hide the authentication path. We also observe that one can trade the disjunctive proof, i.e. a second full evaluation of the hash function, with two comparatively cheap multiplexers.

Our security proof relies on simulation-sound extractactability of the underlying proof system. Hence, on the way to construct ring signatures of sub-linear size, we first prove that Fiat-Shamir-transformed ZKB++ yields a simulation-sound-extractable proof system. For the Unruh-transformed version this result was already known.

One of the questions left open is whether symmetric-key primitives can be used to also construct group signatures, which have similar anonymity features, but the set of users is explicitly managed by a group manager and are also equipped with the possibility to re-identify anonymous signers by a dedicated party. While it is a well-known fact that group signatures following the static security model of Bellare et al. [BMW03] imply public-key encryption [AW04; CG04], one could hope for group signatures in a weaker model. This question was previously pursued by Camenisch et al. [CG04] who presented a construction from one-way functions and non-interactive zero-knowledge arguments. Yet the question whether this construction can be instantiated without structured hardness assumptions remained open. Boneh et al. [BEF19] and Katz et al. [KKW18] answered this question positively for group signatures without opening mechanism.

2.5.2. Double-Authentication-Preventing Signatures

Double-authentication-preventing signatures (DAPS), as introduced by Poettering and Stebila, 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. They provide unforgeability guarantees in the sense of conventional signatures,but have the special property that signing two colliding messages, i.e. message with the same address but differing payloads, allows anybody to extract the secret signing key from the respective signatures. Applications of DAPS include penalizing double-spending attacks in offline transactions of cryptocurrencies [RKS15] or penalizing certification authorities for issuing two certificates with respect to the same domain name, but for two different public keys [BPS17], for example.

Before formally defining DAPS, we first define colliding messages as follows:

Definition 31 (Colliding Messages)**.** We call two messages $m_1 = (a_1, p_1)$ and $m_2 = (a_2, p_2)$ colliding if $a_1 = a_2$, but $p_1 \neq p_2$.

Below, we now formally define DAPS following [PS14; PS17].

Definition 32 (DAPS)**.** A double-authentication-preventing signature scheme DAPS is a tuple $(\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Extract})$ of PPT algorithms, which are defined as follows:

$\mathsf{KGen}(1^\kappa)$: This algorithm takes a security parameter κ as input and outputs a secret (signing) key sk and a public (verification) key pk with associated message space $\mathcal{M}$ (we may omit to make the message space $\mathcal{M}$ explicit).

$\mathsf{Sign}(\mathsf{sk}, m)$: This algorithm takes a secret key sk and a message $m \in \mathcal{M}$ as input and outputs a signature σ.

$\mathsf{Verify}(\mathsf{pk}, m, \sigma)$: This algorithm takes a public key pk, a message $m \in \mathcal{M}$ and a signature σ as input and outputs a bit $b \in \{0, 1\}$.

$\mathsf{Extract}(\mathsf{pk}, m_1, m_2, \sigma_1, \sigma_2)$: This algorithm takes a public key pk, two colliding messages m_1 and m_2 and signatures σ_1 for m_1 and σ_2 for m_2 as inputs and outputs a secret key sk.

Note that the algorithms KGen, Sign, and Verify match the definition of the algorithms of a conventional signature scheme (cf. Definition 26). We also require the same correctness notion as for a conventional signature scheme. For DAPS one requires a restricted but otherwise standard notion of unforgeability [PS14; PS17], where adversaries can adaptively query signatures for messages but only on distinct addresses.

Definition 33 ($\mathsf{EUF\text{-}CMA}$)**.** For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of $\mathsf{EUF\text{-}CMA}$ as

$$\mathsf{Adv}^{\mathsf{daps\text{-}euf\text{-}cma}}_{\mathcal{A},\mathsf{DAPS}}(1^\kappa) = \Pr\left[\mathsf{Exp}^{\mathsf{daps\text{-}euf\text{-}cma}}_{\mathcal{A},\mathsf{DAPS}}(1^\kappa) = 1\right],$$

where the corresponding experiment is depicted in Experiment 11. If for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{daps\text{-}euf\text{-}cma}}_{\mathcal{A},\mathsf{DAPS}}(1^\kappa) \leq \varepsilon(\kappa),$$

we say that DAPS is $\mathsf{EUF\text{-}CMA}$ secure.

Extraction of the secret key from two signatures on colliding messages is ensured by the notion of double-signature extractability (DSE). It requires that, if given signatures on two colliding messages, then the extraction algorithm $\mathsf{Extract}$ recovers the secret key from the two signatures and the public key. We first consider the common notion which requires extraction to work if the key pair has been generated honestly: the adversary is given a key pair and outputs two colliding messages and corresponding signatures. If the signatures are valid and thus satisfy the requirements of the extraction algorithm, yet the key produced by $\mathsf{Extract}$ is different from the signing key, then adversary wins the game.

```
$\mathsf{Exp}^{\mathsf{daps\text{-}euf\text{-}cma}}_{\mathcal{A},\mathsf{DAPS}}(1^\kappa)$:
  $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\kappa)$
  $\mathcal{Q} \leftarrow \emptyset$, $\mathcal{R} \leftarrow \emptyset$
  $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}'(\mathsf{sk},\cdot)}(\mathsf{pk})$
    where oracle $\mathsf{Sign}'(m)$:
      $(a, p) \leftarrow m$
      if $a \in \mathcal{R}$, return $\perp$
      $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$,
      $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$,
      $\mathcal{R} \leftarrow \mathcal{R} \cup \{a\}$
      return $\sigma$
  return 1, if $\mathsf{Verify}(\mathsf{pk}, m^*, \sigma^*) = 1 \ \wedge \ m^* \notin \mathcal{Q}$
  return 0
```

Experiment 11: EUF-CMA security experiment for DAPS.

Definition 34 (DSE)**.** For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of double-signature extraction (DSE) as

$$\mathsf{Adv}^{\mathsf{DSE}}_{\mathcal{A},\mathsf{DAPS}}(1^\kappa) = \Pr\left[\mathsf{Exp}^{\mathsf{DSE}}_{\mathcal{A},\mathsf{DAPS}}(1^\kappa) = 1\right]$$

where the corresponding experiment is depicted in Experiment 12. If for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{DSE}}_{\mathcal{A},\mathsf{DAPS}}(1^\kappa) \leq \varepsilon(\kappa),$$

then DAPS provides DSE.

```
$\mathsf{Exp}^{\mathsf{DSE}}_{\mathcal{A},\mathsf{DAPS}}(1^\kappa)$:
  $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\kappa)$
  $(m_1, m_2, \sigma_1, \sigma_2) \leftarrow \mathcal{A}(\mathsf{sk}, \mathsf{pk})$
  return 0, if $m_1$ and $m_2$ are not colliding
  return 0, if $\mathsf{Verify}(\mathsf{pk}, m_i, \sigma_i) = 0$ for any $i \in [2]$
  $\mathsf{sk}' \leftarrow \mathsf{Extract}(\mathsf{pk}, m_1, m_2, \sigma_1, \sigma_2)$
  return 1, if $\mathsf{sk}' \neq \mathsf{sk}$
  return 0
```

Experiment 12: DSE security experiment for DAPS.

Second, we also recall the strong variant of extractability under malicious keys (denoted as DSE*). In this security experiment, the adversary is allowed to generate the key arbitrarily.

Definition 35 (DSE^*)**.** For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of double-signature extraction under malicious keys (DSE^*) as

$$\mathsf{Adv}^{\mathsf{DSE}^*}_{\mathcal{A},\mathsf{DAPS}}(1^\kappa) = \Pr\left[\mathsf{Exp}^{\mathsf{DSE}^*}_{\mathcal{A},\mathsf{DAPS}}(1^\kappa) = 1\right]$$

where the corresponding experiment is depicted in Experiment 13. If for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{DSE}^*}_{\mathcal{A},\mathsf{DAPS}}(1^\kappa) \leq \varepsilon(\kappa),$$

then DAPS provides DSE^*.

```
$\mathsf{Exp}^{\mathsf{DSE}^*}_{\mathcal{A},\mathsf{DAPS}}(1^\kappa)$:
  $(\mathsf{pk}, m_1, m_2, \sigma_1, \sigma_2) \leftarrow \mathcal{A}(1^\kappa)$
  return 0, if $m_1$ and $m_2$ are not colliding
  return 0, if $\mathsf{Verify}(\mathsf{pk}, m_i, \sigma_i) = 0$ for any $i \in [2]$
  $\mathsf{sk}' \leftarrow \mathsf{Extract}(\mathsf{pk}, m_1, m_2, \sigma_1, \sigma_2)$
  return 1, if $\mathsf{sk}'$ is not the secret key corresponding to $\mathsf{pk}$
  return 0
```

Experiment 13: DSE^* security experiment for DAPS.

We recall one generic construction technique for DAPS based on trapdoor identification schemes. All other prior constructions are ad-hoc.

Trapdoor identification schemes. Trapdoor identification schemes have the additional property, that the prover can sample the commitment in the first round at random, and then—using the trapdoor—can compute the associated randomness. The idea of the double-hash transform [BPS17] is now to first sample the commitment based on the address using a random oracle, and to then perform the identification protocol as normal. Now the verifier can check if the commitment was selected based on the correct address. Once signatures for two colliding messages are obtained, both have been generated with respect to the same commitment, thus the soundness of the identification scheme guarantees the extraction. Bellare et al. [BPS17] also present a second transform, namely the double-ID transform, which follow the same basic idea. This approach works well in the factoring setting using trapdoor identification schemes due to Guillou and Qisquater [GQ88] and Micali and Reyzin [MR02].

This approach exemplifies the close connection between DAPS and identification schemes respectively Σ-protocols. Two signatures σ_1 and σ_2 on colliding messages $m_1 = (a, p_1)$ and $m_2 = (a, p_2)$ have a similarity with transcripts when considering 2-special soundness. There we have (a, e_1, z_1) and (a, e_2, z_2) that share the same first message a. In the case of DAPS, we can view the inputs required to extract as (a, p_1, σ_1) and (a, p_2, σ_2) sharing the same structure as the

Σ-protocol transcripts.

The need for a suitable trapdoor identification scheme limits the applicability of this approach. Both the Guillou-Qisquater and the Micali-Reyzin identification schemes rely on the RSA assumption. Additionally, it is not possible to extend existing signature schemes deployed in practice using this approach. Besides this approach, Boneh et al. [BKN17] present a post-quantum secure construction building on top of public-key encryption schemes with trapdoors from lattices.

As none of the approaches are suitable for extending conventional signature schemes, we follow a different approach in our construction: we extend the signature with a secret share of the secret key. Based on the address, a polynomial of degree 1 is selected for Shamir's secret sharing [Sha79], whereas the payload then determines the share. Then, given two shares from signatures on colliding messages, we obtain two shares with respect to the same polynomial and we are thus able to extract the secret key. To ensure that the signer uses the correct polynomial, i.e. is derived from the address, and that the secret share has been correctly calculated, we add a zero-knowledge proof on to the signature. We explore this approach in two different scenarios. In

- David Derler, Sebastian Ramacher, and Daniel Slamanig. Short Double- and N-Times-Authentication-Preventing Signatures from ECDSA and More. In *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018*, pages 273–287. IEEE, 2018. URL: https://doi.org/10.1109/EuroSP.2018.00027

we first introduce double-signature extractability notions covering the extension from a conventional signature scheme to DAPS. Secondly, we show how our approach can be applied to essentially any signature scheme in the discrete logarithm setting, in particular to ECDSA and Schnorr. Our only requirement on the underlying signature scheme is that the existence of a homomorphism between the group of secret keys and public keys. For schemes in this setting, this is a very natural assumption. However, the size of the address space is limited, since for proving the correct computation of the share, the public key includes encryptions of the coefficients of the used polynomials. Using Shamir's secret sharing also enables us to extend the construction to n-times-authentication-preventing signatures by simply increasing the degree of the sharing polynomial to $n-1$.

One way to fix this short-coming is to first derive the coefficients via an PRF. To ensure that the PRF was evaluated correctly, we prove the correctness of the PRF evaluation. Given a PRF from symmetric-key primitives, we can simply attach a ZKB++-based proof of the correct evaluation. So, we present a construction using a fixes-value-key-binding PRF [CMR98; Fis99] in

- David Derler, Sebastian Ramacher, and Daniel Slamanig. Generic Double-Authentication Preventing Signatures and a Post-quantum Instantiation. In Joonsang Baek, Willy Susilo, and Jongkil Kim, editors, *Provable Security - 12th International Conference, ProvSec 2018, Jeju, South Korea,* *October 25-28, 2018, Proceedings*, volume 11192 of *Lecture Notes in Computer Science*, pages 258–276. Springer, 2018. URL: https://doi.org/10.1007/978-3-030-01446-9_15.

Compared to our work on DL-based instantiation, we obtain a construction with a relaxed requirement on the secret key to public key relation, as we now only require an one-way function. In general, the challenge here is to find an OWF and PRF with compatible domain and codomain combined with an efficient proof system. However, with a focus on our prior work on Picnic, we discuss an instantiation from symmetric-key primitives. Thereby we are able give a partially positive answer to one of the open questions raised by Bellare et al. [BPS17], who asked whether DAPS can be constructed without structured hardness assumptions.

Bibliography

[AAA$^+$19] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, and Yi-Kai Liu. Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process, 2019. URL: https://doi.org/10.6028/NIST.IR.8240.

[AAB$^+$02] Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From Identification to Signatures via the Fiat-Shamir Transform: Minimizing Assumptions for Security and Forward-Security. In Lars R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 418–433. Springer, 2002. URL: https://doi.org/10.1007/3-540-46035-7_28.

[ABB$^+$19] Erdem Alkim, Paulo S. L. M. Barreto, Nina Bindel, Patrick Longa, and Jefferson E. Ricardini. The Lattice-Based Digital Signature Scheme qTESLA. *IACR Cryptology ePrint Archive*, 2019:85, 2019. URL: https://eprint.iacr.org/2019/085.

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient Lattice (H)IBE in the Standard Model. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer, 2010. URL: https://doi.org/10.1007/978-3-642-13190-5_28.

[ABH09] Giuseppe Ateniese, Karyn Benson, and Susan Hohenberger. Key-Private Proxy Re-encryption. In Marc Fischlin, editor, *Topics in Cryptology - CT-RSA 2009, The Cryptographers' Track at the RSA Conference 2009, San Francisco, CA, USA, April 20-24, 2009. Proceedings*, volume 5473 of *Lecture Notes in Computer Science*, pages 279–294. Springer, 2009. URL: https://doi.org/10.1007/978-3-642-00862-7_19.

[Adk11] Heather Adkins. An update on attempted man-in-the-middle attacks. 2011. URL: https://security.googleblog.com/2011/08/update-on-attempted-man-in-middle.html (visited on December 13, 2017).

[AFG$^+$05] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2005, San Diego, California, USA*. The Internet Society, 2005. URL: http://www.isoc.org/isoc/conferences/ndss/05/proceedings/papers/ateniese.pdf.

[AFG$^+$06] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1):1–30, 2006. URL: https://doi.org/10.1145/1127345.1127346.

[AFL$^+$12] Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-Secure Signatures from Lossy Identification Schemes. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 572–590. Springer, 2012. URL: https://doi.org/10.1007/978-3-642-29011-4_34.

[AGO$^+$14] Masayuki Abe, Jens Groth, Miyako Ohkubo, and Mehdi Tibouchi. Structure-Preserving Signatures from Type II Pairings. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 390–407. Springer, 2014. URL: https://doi.org/10.1007/978-3-662-44371-2_22.

[AGR$^+$16] Martin R. Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen. MiMC: Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 191–219, 2016. URL: https://doi.org/10.1007/978-3-662-53887-6_7.

[AHO16] Masayuki Abe, Fumitaka Hoshino, and Miyako Ohkubo. Design in Type-I, Run in Type-III: Fast and Scalable Bilinear-Type Conversion Using Integer Programming. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 387–415. Springer, 2016. URL: https://doi.org/10.1007/978-3-662-53015-3_14.

[Ame04] American Mathematical Society. The Culture of Research and Scholarship in Mathematics: Joint Research and Its Publication. 2004. URL: http://www.ams.org/profession/leaders/culture/CultureStatement04.pdf (visited on January 21, 2019).

[ARS+15] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 430–454. Springer, 2015. URL: https://doi.org/10.1007/978-3-662-46800-5_17.

[ARS+16] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. *IACR Cryptology ePrint Archive*, 2016:687, 2016. URL: http://eprint.iacr.org/2016/687.

[ARU14] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum Attacks on Classical Proof Systems: The Hardness of Quantum Rewinding. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 474–483. IEEE Computer Society, 2014. URL: https://doi.org/10.1109/FOCS.2014.57.

[AW04] Michel Abdalla and Bogdan Warinschi. On the Minimal Assumptions of Group Signature Schemes. In Javier López, Sihan Qing, and Eiji Okamoto, editors, *Information and Communications Security, 6th International Conference, ICICS 2004, Malaga, Spain, October 27-29, 2004, Proceedings*, volume 3269 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2004. URL: https://doi.org/10.1007/978-3-540-30191-2_1.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short Group Signatures. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004. URL: https://doi.org/10.1007/978-3-540-28628-8_3.

[BBS98] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible Protocols and Atomic Proxy Cryptography. In Kaisa Nyberg, editor, *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, volume 1403 of *Lecture Notes in Computer Science*, pages 127–144. Springer, 1998. URL: https://doi.org/10.1007/BFb0054122.

[BD17] Razvan Barbulescu and Sylvain Duquesne. Updating key size estimations for pairings. *IACR Cryptology ePrint Archive*, 2017:334, 2017. URL: http://eprint.iacr.org/2017/334.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random Oracles in a Quantum World. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, 2011. URL: https://doi.org/10.1007/978-3-642-25385-0_3.

[BDH11] Johannes A. Buchmann, Erik Dahmen, and Andreas Hülsing. XMSS - A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings*, volume 7071 of *Lecture Notes in Computer Science*, pages 117–129. Springer, 2011. URL: https://doi.org/10.1007/978-3-642-25405-5_8.

[BDL+12] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. *J. Cryptographic Engineering*, 2(2):77–89, 2012. URL: https://doi.org/10.1007/s13389-012-0027-1.

[BdM93] Josh Cohen Benaloh and Michael de Mare. One-Way Accumulators: A Decentralized Alternative to Digital Sinatures (Extended Abstract). In Tor Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, volume 765 of *Lecture Notes in Computer Science*, pages 274–285. Springer, 1993. URL: https://doi.org/10.1007/3-540-48285-7_24.

[BEF19] Dan Boneh, Saba Eskandarian, and Ben Fisch. Post-quantum EPID Signatures from Symmetric Primitives. In Mitsuru Matsui, editor, *Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4-8, 2019, Proceedings*, volume 11405 of *Lecture Notes in Computer Science*, pages 251–271. Springer, 2019. URL: https://doi.org/10.1007/978-3-030-12612-4_13.

[Ber09] Daniel J. Bernstein. Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete? 2009. URL: https://cr.yp.to/hash/collisioncost-20090823.pdf (visited on March 17, 2019).

[BeS17] BeSafe. Proxy Re-Encryption: Frictionless end-to-end encryption integrated into your workflow. 2017. URL: https://besafe.io/proxy-re-encryption/ (visited on December 14, 2017).

[BF01] Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001. URL: https://doi.org/10.1007/3-540-44647-8_13.

[BHH$^+$15] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. SPHINCS: Practical Stateless Hash-Based Signatures. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 368–397. Springer, 2015. URL: https://doi.org/10.1007/978-3-662-46800-5_15.

[BHT98] Gilles Brassard, Peter HOyer, and Alain Tapp. Quantum Cryptanalysis of Hash and Claw-Free Functions. In Claudio L. Lucchesi and Arnaldo V. Moura, editors, *LATIN '98: Theoretical Informatics, Third Latin American Symposium, Campinas, Brazil, April, 20-24, 1998, Proceedings*, volume 1380 of *Lecture Notes in Computer Science*, pages 163–169. Springer, 1998. URL: https://doi.org/10.1007/BFb0054319.

[BK10] Zvika Brakerski and Yael Tauman Kalai. A Framework for Efficient Signatures, Ring Signatures and Identity Based Encryption in the Standard Model. *IACR Cryptology ePrint Archive*, 2010:86, 2010. URL: http://eprint.iacr.org/2010/086.

[BKM09] Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring Signatures: Stronger Definitions, and Constructions without Random Oracles. *J. Cryptology*, 22(1):114–138, 2009. URL: https://doi.org/10.1007/s00145-007-9011-9.

[BKN17] Dan Boneh, Sam Kim, and Valeria Nikolaenko. Lattice-Based DAPS and Generalizations: Self-enforcement in Signature Schemes. In Dieter Gollmann, Atsuko Miyaji, and Hiroaki Kikuchi, editors, *Applied Cryptography and Network Security - 15th International Conference, ACNS 2017, Kanazawa, Japan, July 10-12, 2017, Proceedings*, volume 10355 of *Lecture Notes in Computer Science*, pages 457–477. Springer, 2017. URL: https://doi.org/10.1007/978-3-319-61204-1_23.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short Signatures from the Weil Pairing. In Colin Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532. Springer, 2001. URL: https://doi.org/10.1007/3-540-45682-1_30.

[BLS02] Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. Constructing Elliptic Curves with Prescribed Embedding Degrees. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *Security in Communication Networks, Third International Conference, SCN 2002, Amalfi, Italy, September 11-13, 2002. Revised Papers*, volume 2576 of *Lecture Notes in Computer Science*, pages 257–267. Springer, 2002. URL: https://doi.org/10.1007/3-540-36413-7_19.

[BM18] Pedro Branco and Paulo Mateus. A Code-Based Linkable Ring Signature Scheme. In Joonsang Baek, Willy Susilo, and Jongkil Kim, editors, *Provable Security - 12th International Conference, ProvSec 2018, Jeju, South Korea, October 25-28, 2018, Proceedings*, volume 11192 of *Lecture Notes in Computer Science*, pages 203–219. Springer, 2018. URL: https://doi.org/10.1007/978-3-030-01446-9_12.

[BM99] Mihir Bellare and Sara K. Miner. A Forward-Secure Digital Signature Scheme. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 431–448. Springer, 1999. URL: https://doi.org/10.1007/3-540-48405-1_28.

[BMW03] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In Eli Biham, editor, *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–629. Springer, 2003. URL: https://doi.org/10.1007/3-540-39200-9_38.

[BN05] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-Friendly Elliptic Curves of Prime Order. In Bart Preneel and Stafford E. Tavares, editors, *Selected Areas in Cryptography, 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers*, volume 3897 of *Lecture Notes in Computer Science*, pages 319–331. Springer, 2005. URL: https://doi.org/10.1007/11693383_22.

[Boy08] Xavier Boyen. The Uber-Assumption Family. In Steven D. Galbraith and Kenneth G. Paterson, editors, *Pairing-Based Cryptography - Pairing 2008, Second International Conference, Egham, UK, September 1-3, 2008. Proceedings*, volume 5209 of *Lecture Notes in Computer Science*, pages 39–56. Springer, 2008. URL: https://doi.org/10.1007/978-3-540-85538-5_3.

[BPS16] Mihir Bellare, Bertram Poettering, and Douglas Stebila. From Identification to Signatures, Tightly: A Framework and Generic Transforms. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 435–464, 2016. URL: https://doi.org/10.1007/978-3-662-53890-6_15.

[BPS17] Mihir Bellare, Bertram Poettering, and Douglas Stebila. Deterring Certificate Subversion: Efficient Double-Authentication-Preventing Signatures. In Serge Fehr, editor, *Public-Key Cryptography - PKC 2017 - 20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, March 28-31, 2017, Proceedings, Part II*, volume 10175 of *Lecture Notes in Computer Science*, pages 121–151. Springer, 2017. URL: https://doi.org/10.1007/978-3-662-54388-7_5.

[BPW12] David Bernhard, Olivier Pereira, and Bogdan Warinschi. How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 626–643. Springer, 2012. URL: https://doi.org/10.1007/978-3-642-34961-4_38.

[BR93] Mihir Bellare and Phillip Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993.* Pages 62–73. ACM, 1993. URL: https://doi.org/10.1145/168588.168596.

[BR96] Mihir Bellare and Phillip Rogaway. The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. In Ueli M. Maurer, editor, *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer, 1996. URL: https://doi.org/10.1007/3-540-68339-9_34.

[Bro05] Daniel R. L. Brown. Generic Groups, Collision Resistance, and ECDSA. *Des. Codes Cryptography*, 35(1):119–152, 2005. URL: http://www.springerlink.com/index/10.1007/s10623-003-6154-z.

[BY03] Mihir Bellare and Bennet S. Yee. Forward-Security in Private-Key Cryptography. In Marc Joye, editor, *Topics in Cryptology - CT-RSA 2003, The Cryptographers' Track at the RSA Conference 2003, San Francisco, CA, USA, April 13-17, 2003, Proceedings*, volume 2612 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2003. URL: https://doi.org/10.1007/3-540-36563-X_1.

[BZ13] Dan Boneh and Mark Zhandry. Secure Signatures and Chosen Ciphertext Security in a Quantum Computing World. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 361–379. Springer, 2013. URL: https://doi.org/10.1007/978-3-642-40084-1_21.

[CCL$^+$14] Nishanth Chandran, Melissa Chase, Feng-Hao Liu, Ryo Nishimaki, and Keita Xagawa. Re-encryption, Functional Re-encryption, and Multi-hop Re-encryption: A Framework for Achieving Obfuscation-Based Security and Instantiations from Lattices. In Hugo Krawczyk, editor, *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings*, volume 8383 of *Lecture Notes in Computer Science*, pages 95–112. Springer, 2014. URL: https://doi.org/10.1007/978-3-642-54631-0_6.

[CG04] Jan Camenisch and Jens Groth. Group Signatures: Better Efficiency and New Theoretical Aspects. In Carlo Blundo and Stelvio Cimato, editors, *Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers*, volume 3352 of *Lecture Notes in Computer Science*, pages 120–133. Springer, 2004. URL: https://doi.org/10.1007/978-3-540-30598-9_9.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004. URL: https://doi.org/10.1145/1008731.1008734.

[CH07] Ran Canetti and Susan Hohenberger. Chosen-ciphertext secure proxy re-encryption. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*, pages 185–194. ACM, 2007. URL: https://doi.org/10.1145/1315245.1315269.

[Cha82] David Chaum. Blind Signatures for Untraceable Payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982.* Pages 199–203. Plenum Press, New York, 1982.

[CHK$^+$10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai Trees, or How to Delegate a Lattice Basis. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552. Springer, 2010. URL: https://doi.org/10.1007/978-3-642-13190-5_27.

[CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A Forward-Secure Public-Key Encryption Scheme. In Eli Biham, editor, *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 255–271. Springer, 2003. URL: https://doi.org/10.1007/3-540-39200-9_16.

[CHR$^+$16] Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. From 5-Pass *MQ* -Based Identification to *MQ* -Based Signatures. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 135–165, 2016. URL: https://doi.org/10.1007/978-3-662-53890-6_5.

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 56–72. Springer, 2004. URL: https://doi.org/10.1007/978-3-540-28628-8_4.

[CMR98] Ran Canetti, Daniele Micciancio, and Omer Reingold. Perfectly One-Way Probabilistic Hash Functions (Preliminary Version). In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 131–140. ACM, 1998. URL: https://doi.org/10.1145/276698.276721.

[Cry] CryptoNote. CrytpoNote Phylosopyhy. URL: https://cryptonote.org/inside/ (visited on March 26, 2019).

[CSF+08] David Cooper, Stefan Santesson, Stephen Farrell, Sharon Boeyen, Russell Housley, and W. Timothy Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. *RFC*, 5280:1–151, 2008. URL: https://doi.org/10.17487/RFC5280.

[CvH91] David Chaum and Eugène van Heyst. Group Signatures. In Donald W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer, 1991. URL: https://doi.org/10.1007/3-540-46416-6_22.

[Dam10] Ivan Damgård. On Σ-protocols. 2010. URL: http://www.cs.au.dk/~ivan/Sigma.pdf (visited on October 12, 2017).

[DEG+18] Christoph Dobraunig, Maria Eichlseder, Lorenzo Grassi, Virginie Lallemand, Gregor Leander, Eik List, Florian Mendel, and Christian Rechberger. Rasta: A Cipher with Low ANDdepth and Few ANDs per Bit. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 662–692. Springer, 2018. URL: https://doi.org/10.1007/978-3-319-96884-1_22.

[DEM15] Christoph Dobraunig, Maria Eichlseder, and Florian Mendel. Higher-Order Cryptanalysis of LowMC. In Soonhak Kwon and Aaram Yun, editors, *Information Security and Cryptology - ICISC 2015 - 18th International Conference, Seoul, South Korea, November 25-27, 2015, Revised Selected Papers*, volume 9558 of *Lecture Notes in Computer Science*, pages 87–101. Springer, 2015. URL: https://doi.org/10.1007/978-3-319-30840-1_6.

[DFM+19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model. *IACR Cryptology ePrint Archive*, 2019:190, 2019. URL: https://eprint.iacr.org/2019/190.

[DGV+16] Özgür Dagdelen, David Galindo, Pascal Véron, Sidi Mohamed El Yousfi Alaoui, and Pierre-Louis Cayrel. Extended security arguments for signature schemes. *Des. Codes Cryptography*, 78(2):441–461, 2016. URL: https://doi.org/10.1007/s10623-014-0009-7.

[DHS15] David Derler, Christian Hanser, and Daniel Slamanig. Revisiting Cryptographic Accumulators, Additional Properties and Relations to Other Primitives. In Kaisa Nyberg, editor, *Topics in Cryptology - CT-RSA 2015, The Cryptographer's Track at the RSA Conference 2015, San Francisco, CA, USA, April 20-24, 2015. Proceedings*, volume 9048 of *Lecture Notes in Computer Science*, pages 127–144. Springer, 2015. URL: https://doi.org/10.1007/978-3-319-16715-2_7.

[Din18] Itai Dinur. Linear Equivalence of Block Ciphers with Partial Non-Linear Layers: Application to LowMC. *IACR Cryptology ePrint Archive*, 2018:772, 2018. URL: https://eprint.iacr.org/2018/772.

[DJS$^+$18] David Derler, Tibor Jager, Daniel Slamanig, and Christoph Striecks. Bloom Filter Encryption and Applications to Efficient Forward-Secret 0-RTT Key Exchange. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 425–455. Springer, 2018. URL: https://doi.org/10.1007/978-3-319-78372-7_14.

[DKL$^+$18b] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):238–268, 2018. URL: https://doi.org/10.13154/tches.v2018.i1.238-268.

[DKN$^+$04] Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous Identification in Ad Hoc Groups. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 609–626. Springer, 2004. URL: https://doi.org/10.1007/978-3-540-24676-3_36.

[DLM$^+$15] Itai Dinur, Yunwen Liu, Willi Meier, and Qingju Wang. Optimized Interpolation Attacks on LowMC. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 535–560. Springer, 2015. URL: https://doi.org/10.1007/978-3-662-48800-3_22.

[DS16] David Derler and Daniel Slamanig. Key-Homomorphic Signatures and Applications to Multiparty Signatures. *IACR Cryptology ePrint Archive*, 2016:792, 2016. URL: http://eprint.iacr.org/2016/792.

[DSS05] C. Dods, Nigel P. Smart, and Martijn Stam. Hash Based Digital Signature Schemes. In Nigel P. Smart, editor, *Cryptography and Coding, 10th IMA International Conference, Cirencester, UK, December 19-21, 2005, Proceedings*, volume 3796 of *Lecture Notes* *in Computer Science*, pages 96–115. Springer, 2005. URL: https://doi.org/10.1007/11586821_8.

[EJ17] Nadia El Mrabet and Marc Joye. *Guide to pairing-based cryptography*. Chapman and Hall/CRC, 2017.

[Fis18] Dennis Fisher. 500 Million Affected in Marriott Data Breach. 2018. URL: https://duo.com/decipher/500-million-affected-in-marriott-data-breach (visited on April 10, 2019).

[Fis99] Marc Fischlin. Pseudorandom Function Tribe Ensembles Based on One-Way Permutations: Improvements and Applications. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 432–445. Springer, 1999. URL: https://doi.org/10.1007/3-540-48910-X_30.

[FJP14] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *J. Mathematical Cryptology*, 8(3):209–247, 2014. URL: https://doi.org/10.1515/jmc-2012-0015.

[FKM$^+$12] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the Non-malleability of the Fiat-Shamir Transform. In Steven D. Galbraith and Mridul Nandi, editors, *Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings*, volume 7668 of *Lecture Notes in Computer Science*, pages 60–79. Springer, 2012. URL: https://doi.org/10.1007/978-3-642-34931-7_5.

[FKP16] Manuel Fersch, Eike Kiltz, and Bertram Poettering. On the Provable Security of (EC)DSA Signatures. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1651–1662. ACM, 2016. URL: https://doi.org/10.1145/2976749.2978413.

[FLR$^+$10] Marc Fischlin, Anja Lehmann, Thomas Ristenpart, Thomas Shrimpton, Martijn Stam, and Stefano Tessaro. Random Oracles with(out) Programmability. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 303–320. Springer, 2010. URL: https://doi.org/10.1007/978-3-642-17373-8_18.

[FR94] Gerhard Frey and Hans-Georg Rück. A Remark Concerning m-Divisibility and the Discrete Logarithm in the Divisor Class Group of Curves. *Mathematics of Computation*, 62(206):865–874, 1994. ISSN: 00255718. URL: http://www.jstor.org/stable/2153546.

[FS07] Eiichiro Fujisaki and Koutarou Suzuki. Traceable Ring Signature. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography - PKC 2007, 10th International Conference on Practice and Theory in Public-Key Cryptography, Beijing, China, April 16-20, 2007, Proceedings*, volume 4450 of *Lecture Notes in Computer Science*, pages 181–200. Springer, 2007. URL: https://doi.org/10.1007/978-3-540-71677-8_13.

[FS86] Amos Fiat and Adi Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986. URL: https://doi.org/10.1007/3-540-47721-7_12.

[GA07] Matthew Green and Giuseppe Ateniese. Identity-Based Proxy Re-encryption. In Jonathan Katz and Moti Yung, editors, *Applied Cryptography and Network Security, 5th International Conference, ACNS 2007, Zhuhai, China, June 5-8, 2007, Proceedings*, volume 4521 of *Lecture Notes in Computer Science*, pages 288–306. Springer, 2007. URL: https://doi.org/10.1007/978-3-540-72738-5_19.

[Gam84] Taher El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer, 1984. URL: https://doi.org/10.1007/3-540-39568-7_2.

[GCZ16] Steven Goldfeder, Melissa Chase, and Greg Zaverucha. Efficient Post-Quantum Zero-Knowledge and Signatures. *IACR Cryptology ePrint Archive*, 2016:1110, 2016. URL: http://eprint.iacr.org/2016/1110.

[Gha16] Essam Ghadafi. Short Structure-Preserving Signatures. In Kazue Sako, editor, *Topics in Cryptology - CT-RSA 2016 - The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29 - March 4, 2016, Proceedings*, volume 9610 of *Lecture Notes in Computer Science*, pages 305–321. Springer, 2016. URL: https://doi.org/10.1007/978-3-319-29485-8_18.

[GHJ+17] Felix Günther, Britta Hale, Tibor Jager, and Sebastian Lauer. 0-RTT Key Exchange with Full Forward Secrecy. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III*, volume 10212 of *Lecture Notes in Computer Science*, pages 519–548, 2017. URL: https://doi.org/10.1007/978-3-319-56617-7_18.

[GI14] Niv Gilboa and Yuval Ishai. Distributed Point Functions and Their Applications. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 640–658. Springer, 2014. URL: https://doi.org/10.1007/978-3-642-55220-5_35.

[GM15] Matthew D. Green and Ian Miers. Forward Secure Asynchronous Messaging from Puncturable Encryption. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pages 305–320. IEEE Computer Society, 2015. URL: https://doi.org/10.1109/SP.2015.26.

[GMO16] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. ZKBoo: Faster Zero-Knowledge for Boolean Circuits. In Thorsten Holz and Stefan Savage, editors, *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.* Pages 1069–1083. USENIX Association, 2016. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/giacomelli.

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that Yield Nothing But their Validity and a Methodology of Cryptographic Protocol Design (Extended Abstract). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 174–187. IEEE Computer Society, 1986. URL: https://doi.org/10.1109/SFCS.1986.47.

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001. ISBN: 0-521-79172-3.

[Gol04] Oded Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004. ISBN: 0-521-83084-2.

[Gol08] Oded Goldreich. *Computational complexity - a conceptual perspective*. Cambridge University Press, 2008. ISBN: 978-0-521-88473-0.

[Gol10] Oded Goldreich. *P, NP, and NP-Completeness: The Basics of Complexity Theory*. Cambridge University Press, 2010. ISBN: 978-0-521-12254-2.

[GPS17] Steven D. Galbraith, Christophe Petit, and Javier Silva. Identification Protocols and Signature Schemes Based on Supersingular Isogeny Problems. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 3–33. Springer, 2017. URL: https://doi.org/10.1007/978-3-319-70694-8_1.

[GQ88] Louis C. Guillou and Jean-Jacques Quisquater. A ”Paradoxical” Indentity-Based Signature Scheme Resulting from Zero-Knowledge. In Shafi Goldwasser, editor, *Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings*, volume 403 of *Lecture Notes in Computer Science*, pages 216–231. Springer, 1988. URL: https://doi.org/10.1007/0-387-34799-2_16.

[Gro96] Lov K. Grover. A Fast Quantum Mechanical Algorithm for Database Search. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219. ACM, 1996. URL: https://doi.org/10.1145/237814.237866.

[GRR+16] Lorenzo Grassi, Christian Rechberger, Dragos Rotaru, Peter Scholl, and Nigel P. Smart. MPC-Friendly Symmetric Key Primitives. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 430–443. ACM, 2016. URL: https://doi.org/10.1145/2976749.2978332.

[Gün89] Christoph G. Günther. An Identity-Based Key-Exchange Protocol. In Jean-Jacques Quisquater and Joos Vandewalle, editors, *Advances in Cryptology - EUROCRYPT '89, Workshop on the Theory and Application of of Cryptographic Techniques, Houthalen, Belgium, April 10-13, 1989, Proceedings*, volume 434 of *Lecture Notes in Computer Science*, pages 29–37. Springer, 1989. URL: https://doi.org/10.1007/3-540-46885-4_5.

[HBG+18] Andreas Hülsing, Denis Butin, Stefan-Lukas Gazdag, Joost Rijneveld, and Aziz Mohaisen. XMSS: eXtended Merkle Signature Scheme. *RFC*, 8391:1–74, 2018. URL: https://doi.org/10.17487/RFC8391.

[HL08] Carmit Hazay and Yehuda Lindell. Efficient Protocols for Set Intersection and Pattern Matching with Security Against Malicious and Covert Adversaries. In Ran Canetti, editor, *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008.* Volume 4948 of *Lecture Notes in Computer Science*, pages 155–175. Springer, 2008. URL: `https://doi.org/10.1007/978-3-540-78524-8_10`.

[Imp95] Russell Impagliazzo. A Personal View of Average-Case Complexity. In *Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, June 19-22, 1995*, pages 134–147. IEEE Computer Society, 1995. URL: `https://doi.org/10.1109/SCT.1995.514853`.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the Provable Consequences of One-Way Permutations. In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 44–61. ACM, 1989. URL: `https://doi.org/10.1145/73007.73012`.

[IY87] Russell Impagliazzo and Moti Yung. Direct Minimum-Knowledge Computations. In Carl Pomerance, editor, *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*, volume 293 of *Lecture Notes in Computer Science*, pages 40–51. Springer, 1987. URL: `https://doi.org/10.1007/3-540-48184-2_4`.

[JF11] David Jao and Luca De Feo. Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings*, volume 7071 of *Lecture Notes in Computer Science*, pages 19–34. Springer, 2011. URL: `https://doi.org/10.1007/978-3-642-25405-5_2`.

[JK03] Jakob Jonsson and Burt Kaliski. Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1. *RFC*, 3447:1–72, 2003. URL: `https://doi.org/10.17487/RFC3447`.

[Jou00] Antoine Joux. A One Round Protocol for Tripartite Diffie-Hellman. In Wieb Bosma, editor, *Algorithmic Number Theory, 4th International Symposium, ANTS-IV, Leiden, The Netherlands, July 2-7, 2000, Proceedings*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394. Springer, 2000. URL: `https://doi.org/10.1007/10722028_23`.

[Kat10] Jonathan Katz. *Digital Signatures.* Springer, 2010. ISBN: 978-0-387-27711-0. URL: `https://doi.org/10.1007/978-0-387-27712-7`.

[KKW18] Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. Improved Non-Interactive Zero Knowledge with Applications to Post-Quantum Signatures. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, pages 525–537. ACM, 2018. URL: https://doi.org/10.1145/3243734.3243805.

[KL14] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition.* CRC Press, 2014. ISBN: 9781466570269.

[KLS18] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A Concrete Treatment of Fiat-Shamir Signatures in the Quantum Random-Oracle Model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 552–586. Springer, 2018. URL: https://doi.org/10.1007/978-3-319-78372-7_18.

[KM07] Neal Koblitz and Alfred Menezes. Another look at generic groups. *Adv. in Math. of Comm.*, 1(1):13–28, 2007. URL: https://doi.org/10.3934/amc.2007.1.13.

[KM15] Neal Koblitz and Alfred J. Menezes. The random oracle model: a twenty-year retrospective. *Des. Codes Cryptography*, 77(2-3):587–610, 2015. URL: https://doi.org/10.1007/s10623-015-0094-2.

[KMP16] Eike Kiltz, Daniel Masny, and Jiaxin Pan. Optimal Security Proofs for Signatures from Identification Schemes. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 33–61. Springer, 2016. URL: https://doi.org/10.1007/978-3-662-53008-5_2.

[KR00] Hugo Krawczyk and Tal Rabin. Chameleon Signatures. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2000, San Diego, California, USA*. The Internet Society, 2000. URL: http://www.isoc.org/isoc/conferences/ndss/2000/proceedings/042.pdf.

[KSD13] Cameron F. Kerry, Acting Secretary, and Charles Romine Director. Digital Signature Standard (DSS). National Institute of Standards and Technology (NIST), FIPS PUB 186, U.S. Department of Commerce, 2013.

[KSS08] Ezekiel J. Kachisa, Edward F. Schaefer, and Michael Scott. Constructing Brezing-Weng Pairing-Friendly Elliptic Curves Using Elements in the Cyclotomic Field. In Steven D. Galbraith and Kenneth G. Paterson, editors, *Pairing-Based Cryptography - Pairing 2008, Second International Conference, Egham, UK, September 1-3, 2008. Proceedings*, volume 5209 of *Lecture Notes in Computer Science*, pages 126–135. Springer, 2008. URL: https://doi.org/10.1007/978-3-540-85538-5_9.

[KW03] Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, *Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS 2003, Washington, DC, USA, October 27-30, 2003*, pages 155–164. ACM, 2003. URL: https://doi.org/10.1145/948109.948132.

[Lam79] Leslie Lamport. Constructing digital signatures from one-way functions. Technical report, SRI Intl. Computer Science Laboratory, 1979.

[Lau14] Ben Laurie. Certificate Transparency. *ACM Queue*, 12(8):10–19, 2014. URL: https://doi.org/10.1145/2668152.2668154.

[LFK+90] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic Methods for Interactive Proof Systems. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 2–10. IEEE Computer Society, 1990. URL: https://doi.org/10.1109/FSCS.1990.89518.

[LG15] Wouter Lueks and Ian Goldberg. Sublinear Scaling for Multi-Client Private Information Retrieval. In Rainer Böhme and Tatsuaki Okamoto, editors, *Financial Cryptography and Data Security - 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers*, volume 8975 of *Lecture Notes in Computer Science*, pages 168–186. Springer, 2015. URL: https://doi.org/10.1007/978-3-662-47854-7_10.

[Lic69] Stephen Lichtenbaum. Duality theorems for curves over P-adic fields. *Inventiones mathematicae*, 7(2):120–136, 1969. ISSN: 0020-9910. URL: http://dx.doi.org/10.1007/BF01389795.

[LLK13] Ben Laurie, Adam Langley, and Emilia Käsper. Certificate Transparency. *RFC*, 6962:1–27, 2013. URL: https://doi.org/10.17487/RFC6962.

[LLN+16] Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Zero-Knowledge Arguments for Lattice-Based Accumulators: Logarithmic-Size Ring Signatures and Group Signatures Without Trapdoors. In Marc Fischlin and Jean-Sébastien Coron, editors, *Ad-*

vances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II, volume 9666 of *Lecture Notes in Computer Science*, pages 1–31. Springer, 2016. URL: https://doi.org/10.1007/978-3-662-49896-5_1.

[LM09] Vadim Lyubashevsky and Daniele Micciancio. On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 577–594. Springer, 2009. URL: https://doi.org/10.1007/978-3-642-03356-8_34.

[LV08a] Benoît Libert and Damien Vergnaud. Tracing Malicious Proxies in Proxy Re-encryption. In Steven D. Galbraith and Kenneth G. Paterson, editors, *Pairing-Based Cryptography - Pairing 2008, Second International Conference, Egham, UK, September 1-3, 2008. Proceedings*, volume 5209 of *Lecture Notes in Computer Science*, pages 332–353. Springer, 2008. URL: https://doi.org/10.1007/978-3-540-85538-5_22.

[LV08b] Benoît Libert and Damien Vergnaud. Unidirectional Chosen-Ciphertext Secure Proxy Re-encryption. In Ronald Cramer, editor, *Public Key Cryptography - PKC 2008, 11th International Workshop on Practice and Theory in Public-Key Cryptography, Barcelona, Spain, March 9-12, 2008. Proceedings*, volume 4939 of *Lecture Notes in Computer Science*, pages 360–379. Springer, 2008. URL: https://doi.org/10.1007/978-3-540-78440-1_21.

[LV11] Benoît Libert and Damien Vergnaud. Unidirectional Chosen-Ciphertext Secure Proxy Re-Encryption. *IEEE Trans. Information Theory*, 57(3):1786–1802, 2011. URL: https://doi.org/10.1109/TIT.2011.2104470.

[Lyu09] Vadim Lyubashevsky. Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In Mitsuru Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, volume 5912 of *Lecture Notes in Computer Science*, pages 598–616. Springer, 2009. URL: https://doi.org/10.1007/978-3-642-10366-7_35.

[LZ19] Qipeng Liu and Mark Zhandry. Revisiting Post-Quantum Fiat-Shamir. *IACR Cryptology ePrint Archive*, 2019:262, 2019. URL: https://eprint.iacr.org/2019/262.

[McE78] Robert J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. *Coding Thv*, 4244:114–116, 1978.

[Mer89] Ralph C. Merkle. A Certified Digital Signature. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 218–238. Springer, 1989. URL: https://doi.org/10.1007/0-387-34805-0_21.

[MI88] Tsutomu Matsumoto and Hideki Imai. Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption. In Christoph G. Günther, editor, *Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings*, volume 330 of *Lecture Notes in Computer Science*, pages 419–453. Springer, 1988. URL: https://doi.org/10.1007/3-540-45961-8_39.

[Mil04] Victor S. Miller. The Weil Pairing, and Its Efficient Calculation. *J. Cryptology*, 17(4):235–261, 2004. URL: https://doi.org/10.1007/s00145-004-0315-8.

[Mil86] Victor S. Miller. Short Programs for functions on Curves. In *IBM Thomas J. Watson Resarch Center*, 1986. unpublished.

[Mim17] Michael Mimoso. Adobe Private PGP Key Leak a Blunder, But It Could Have Been Worse. 2017. URL: https://threatpost.com/adobe-private-pgp-key-leak-a-blunder-but-it-could-have-been-worse/128113/ (visited on December 14, 2017).

[MKF$^+$16] David A. McGrew, Panos Kampanakis, Scott R. Fluhrer, Stefan-Lukas Gazdag, Denis Butin, and Johannes A. Buchmann. State Management for Hash-Based Signatures. In Lidong Chen, David A. McGrew, and Chris J. Mitchell, editors, *Security Standardisation Research - Third International Conference, SSR 2016, Gaithersburg, MD, USA, December 5-6, 2016, Proceedings*, volume 10074 of *Lecture Notes in Computer Science*, pages 244–260. Springer, 2016. URL: https://doi.org/10.1007/978-3-319-49100-4_11.

[MN17] Bart Mennink and Samuel Neves. Optimal PRFs from Blockcipher Designs. *IACR Trans. Symmetric Cryptol.*, 2017(3):228–252, 2017. URL: https://doi.org/10.13154/tosc.v2017.i3.228-252.

[Mon] Moneropedia. Ring Signature. URL: https://ww.getmonero.org/resources/moneropedia/ringsignatures.html (visited on March 26, 2019).

[MR02] Silvio Micali and Leonid Reyzin. Improving the Exact Security of Digital Signature Schemes. *J. Cryptology*, 15(1):1–18, 2002. URL: https://doi.org/10.1007/s00145-001-0005-8.

[MS02] John Malone-Lee and Nigel P. Smart. Modifications of ECDSA. In Kaisa Nyberg and Howard M. Heys, editors, *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002, St. John's, Newfoundland, Canada, August 15-16, 2002. Revised Papers*, volume 2595 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2002. URL: https://doi.org/10.1007/3-540-36492-7_1.

[MVO91] Alfred Menezes, Scott A. Vanstone, and Tatsuaki Okamoto. Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field. In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 80–89. ACM, 1991. URL: https://doi.org/10.1145/103418.103434.

[Nie86] Harald Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986.

[NK95] Kaisa Nyberg and Lars R. Knudsen. Provable Security Against a Differential Attack. *J. Cryptology*, 8(1):27–37, 1995. URL: https://doi.org/10.1007/BF00204800.

[Noe15] Shen Noether. Ring Signature Confidential Transactions for Monero. *IACR Cryptology ePrint Archive*, 2015:1098, 2015. URL: http://eprint.iacr.org/2015/1098.

[Nuñ18] David Nuñez. Umbral: a threshold proxy re-encryption scheme. 2018. URL: https://github.com/nucypher/umbral-doc/raw/master/umbral-doc.pdf (visited on February 7, 2019).

[OO98] Kazuo Ohta and Tatsuaki Okamoto. On Concrete Security Treatment of Signatures Derived from Identification. In Hugo Krawczyk, editor, *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, volume 1462 of *Lecture Notes in Computer Science*, pages 354–369. Springer, 1998. URL: https://doi.org/10.1007/BFb0055741.

[Phi11] Phillip. Comodo SSL Affiliate The Recent RA Compromise. 2011. URL: https://blog.comodo.com/other/the-recent-ra-compromise/ (visited on December 13, 2017).

[Pol15] David Politis. Google Apps vs Office 365: Comparing the Usage, Adoption, and Effectiveness of Cloud IT's Power Players. 2015. URL: https://www.bettercloud.com/monitor/google-apps-vs-office-365/ (visited on March 12, 2019).

[PRS$^+$17] Yuriy Polyakov, Kurt Rohloff, Gyana Sahu, and Vinod Vaikuntanathan. Fast Proxy Re-Encryption for Publish/Subscribe Systems. *ACM Trans. Priv. Secur.*, 20(4):14:1–14:31, 2017. URL: https://doi.org/10.1145/3128607.

[PS00] David Pointcheval and Jacques Stern. Security Arguments for Digital Signatures and Blind Signatures. *J. Cryptology*, 13(3):361–396, 2000. URL: https://doi.org/10.1007/s001450010003.

[PS14] Bertram Poettering and Douglas Stebila. Double-Authentication-Preventing Signatures. In Miroslaw Kutylowski and Jaideep Vaidya, editors, *Computer Security - ESORICS 2014 - 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part I*, volume 8712 of *Lecture Notes in Computer Science*, pages 436–453. Springer, 2014. URL: https://doi.org/10.1007/978-3-319-11203-9_25.

[PS16] David Pointcheval and Olivier Sanders. Short Randomizable Signatures. In Kazue Sako, editor, *Topics in Cryptology - CT-RSA 2016 - The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29 - March 4, 2016, Proceedings*, volume 9610 of *Lecture Notes in Computer Science*, pages 111–126. Springer, 2016. URL: https://doi.org/10.1007/978-3-319-29485-8_7.

[PS17] Bertram Poettering and Douglas Stebila. Double-authentication-preventing signatures. *Int. J. Inf. Sec.*, 16(1):1–22, 2017. URL: https://doi.org/10.1007/s10207-015-0307-8.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive Zero Knowledge for NP from (Plain) Learning With Errors. *IACR Cryptology ePrint Archive*, 2019:158, 2019. URL: https://eprint.iacr.org/2019/158.

[PS96] David Pointcheval and Jacques Stern. Security Proofs for Signature Schemes. In Ueli M. Maurer, editor, *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398. Springer, 1996. URL: https://doi.org/10.1007/3-540-68339-9_33.

[Reg06] Oded Regev. Lattice-Based Cryptography. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 131–141. Springer, 2006. URL: https://doi.org/10.1007/11818175_8.

[Res18] Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. *RFC*, 8446:1–160, 2018. URL: https://doi.org/10.17487/RFC8446.

[RGW$^+$10] Yanli Ren, Dawu Gu, Shuozhong Wang, and Xinpeng Zhang. Hierarchical Identity-Based Proxy Re-Encryption without Random Oracles. *Int. J. Found. Comput. Sci.*, 21(6):1049–1063, 2010. URL: https://doi.org/10.1142/S0129054110007726.

[RKS15] Tim Ruffing, Aniket Kate, and Dominique Schröder. Liar, Liar, Coins on Fire!: Penalizing Equivocation By Loss of Bitcoins. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 219–230. ACM, 2015. URL: https://doi.org/10.1145/2810103.2813686.

[RSS17] Dragos Rotaru, Nigel P. Smart, and Martijn Stam. Modes of Operation Suitable for Computing on Encrypted Data. *IACR Trans. Symmetric Cryptol.*, 2017(3):294–324, 2017. URL: https://doi.org/10.13154/tosc.v2017.i3.294-324.

[RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to Leak a Secret. In Colin Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001. URL: https://doi.org/10.1007/3-540-45682-1_32.

[RST18] Christian Rechberger, Hadi Soleimany, and Tyge Tiessen. Cryptanalysis of Low-Data Instances of Full LowMCv2. *IACR Trans. Symmetric Cryptol.*, 2018(3):163–181, 2018. URL: https://doi.org/10.13154/tosc.v2018.i3.163-181.

[Sae18] Brennan Saeta. Cloud TPU now offers preemptible pricing and global availability. 2018. URL: https://cloudplatform.googleblog.com/2018/06/Cloud-TPU-now-offers-preemptible-pricing-and-global-availability.html (visited on March 26, 2019).

[SAL$^+$17] Shifeng Sun, Man Ho Au, Joseph K. Liu, and Tsz Hon Yuen. RingCT 2.0: A Compact Accumulator-Based (Linkable Ring Signature) Protocol for Blockchain Cryptocurrency Monero. In Simon N. Foley, Dieter Gollmann, and Einar Snekkenes, editors, *Computer Security - ESORICS 2017 - 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II*, volume 10493 of *Lecture Notes in Computer Science*, pages 456–474. Springer, 2017. URL: https://doi.org/10.1007/978-3-319-66399-9_25.

[Sch18] Emily Schechter. A secure web is here to stay. 2018. URL: https://security.googleblog.com/2018/02/a-secure-web-is-here-to-stay.html (visited on February 13, 2018).

[Sch19] Berry Schoenmakers. Lecture Notes Cryptographic Protocols. 2019. URL: https://www.win.tue.nl/~berry/CryptographicProtocols/LectureNotes.pdf (visited on March 3, 2019).

[Sch89] Claus-Peter Schnorr. Efficient Identification and Signatures for Smart Cards. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 1989. URL: https://doi.org/10.1007/0-387-34805-0_22.

[Sha79] Adi Shamir. How to Share a Secret. *Commun. ACM*, 22(11):612–613, 1979. URL: http://doi.acm.org/10.1145/359168.359176.

[Sha84] Adi Shamir. Identity-Based Cryptosystems and Signature Schemes. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984. URL: https://doi.org/10.1007/3-540-39568-7_5.

[Sha90] Adi Shamir. IP=PSPACE. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 11–15. IEEE Computer Society, 1990. URL: https://doi.org/10.1109/FSCS.1990.89519.

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. *IACR Cryptology ePrint Archive*, 2004:332, 2004. URL: http://eprint.iacr.org/2004/332.

[Sho94] Peter W. Shor. Polynominal time algorithms for discrete logarithms and factoring on a quantum computer. In Leonard M. Adleman and Ming-Deh A. Huang, editors, *Algorithmic Number Theory, First International Symposium, ANTS-I, Ithaca, NY, USA, May 6-9, 1994, Proceedings*, volume 877 of *Lecture Notes in Computer Science*, page 289. Springer, 1994. URL: https://doi.org/10.1007/3-540-58691-1_68.

[Sho97] Victor Shoup. Lower Bounds for Discrete Logarithms and Related Problems. In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer, 1997. URL: https://doi.org/10.1007/3-540-69053-0_18.

[Sma16] Nigel P. Smart. *Cryptography Made Simple*. Information Security and Cryptography. Springer, 2016. ISBN: 978-3-319-21935-6. URL: https://doi.org/10.1007/978-3-319-21936-3.

[SSH11] Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. Public-Key Identification Schemes Based on Multivariate Quadratic Polynomials. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of

Lecture Notes in Computer Science, pages 706–723. Springer, 2011. URL: https://doi.org/10.1007/978-3-642-22792-9_40.

[Sta19] Internet World Stats. Internet growth statistics. 2019. URL: https://www.internetworldstats.com/emarketing.htm (visited on March 13, 2019).

[Ste18] John Stevens. Internet Stats & Facts for 2019. 2018. URL: https://hostingfacts.com/internet-facts-stats/ (visited on April 6, 2019).

[Ste93] Jacques Stern. A New Identification Scheme Based on Syndrome Decoding. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 13–21. Springer, 1993. URL: https://doi.org/10.1007/3-540-48329-2_2.

[Sul17] Nick Sullivan. https://blog.cloudflare.com/geo-key-manager-how-it-works/. 2017. URL: https://blog.cloudflare.com/geo-key-manager-how-it-works/ (visited on December 13, 2017).

[Tan08] Qiang Tang. Type-Based Proxy Re-encryption and Its Construction. In Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *Progress in Cryptology - INDOCRYPT 2008, 9th International Conference on Cryptology in India, Kharagpur, India, December 14-17, 2008. Proceedings*, volume 5365 of *Lecture Notes in Computer Science*, pages 130–144. Springer, 2008. URL: https://doi.org/10.1007/978-3-540-89754-5_11.

[Tat57] John Tate. WC-groups over p-adic fields. eng. *Séminaire Bourbaki*, 4:265–277, 1957. URL: http://eudml.org/doc/109548.

[Tat63] John Tate. Duality theorems in Galois cohomology over number fields. Proc. Int. Congr. Math. 1962, 288-295 (1963). 1963.

[TU16] Ehsan Ebrahimi Targhi and Dominique Unruh. Post-Quantum Security of the Fujisaki-Okamoto and OAEP Transforms. In Martin Hirt and Adam D. Smith, editors, *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, volume 9986 of *Lecture Notes in Computer Science*, pages 192–216, 2016. URL: https://doi.org/10.1007/978-3-662-53644-5_8.

[Unr15] Dominique Unruh. Non-Interactive Zero-Knowledge Proofs in the Quantum Random Oracle Model. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 755–784. Springer, 2015. URL: https://doi.org/10.1007/978-3-662-46803-6_25.

[Unr17] Dominique Unruh. Post-quantum Security of Fiat-Shamir. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 65–95. Springer, 2017. URL: https://doi.org/10.1007/978-3-319-70694-8_3.

[Vér96] Pascal Véron. Improved identification schemes based on error-correcting codes. *Appl. Algebra Eng. Commun. Comput.*, 8(1):57–69, 1996. URL: https://doi.org/10.1007/s002000050053.

[VP17] Mathy Vanhoef and Frank Piessens. Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1313–1328. ACM, 2017. URL: https://doi.org/10.1145/3133956.3134027.

[vSMJ+12] Nicolas van Saberhagen, Johannes Meier, Antonio M. Juarez, Max Jameson, and Seigen. CryptoNote Signatures. 2012. URL: https://cryptonote.org/cns/cns002.txt (visited on March 26, 2019).

[Wat05] Brent Waters. Efficient Identity-Based Encryption Without Random Oracles. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2005. URL: https://doi.org/10.1007/11426639_7.

[Wei40] André Weil. Sur les fonctions algébriques à corps de constantes fini. In Les Comptes rendus de l'Académie des sciences, pages 592–594, 1940.

[WYT+09] Jian Weng, Yanjiang Yang, Qiang Tang, Robert H. Deng, and Feng Bao. Efficient Conditional Proxy Re-encryption with Chosen-Ciphertext Security. In Pierangela Samarati, Moti Yung, Fabio Martinelli, and Claudio Agostino Ardagna, editors, *Information Security, 12th International Conference, ISC 2009, Pisa, Italy, September 7-9, 2009. Proceedings*, volume 5735 of *Lecture Notes in Computer Science*, pages 151–166. Springer, 2009. URL: https://doi.org/10.1007/978-3-642-04474-8_13.

[YAJ+17] Youngho Yoo, Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev. A Post-quantum Digital Signature Scheme Based on Supersingular Isogenies. In Aggelos Kiayias, editor, *Financial Cryptography and Data Security - 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Se-*

lected Papers, volume 10322 of *Lecture Notes in Computer Science*, pages 163–181. Springer, 2017. URL: https://doi.org/10.1007/978-3-319-70972-7_9.

[Yao86] Andrew Chi-Chih Yao. How to Generate and Exchange Secrets (Extended Abstract). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 162–167. IEEE Computer Society, 1986. URL: https://doi.org/10.1109/SFCS.1986.25.

[Zha12] Mark Zhandry. Secure Identity-Based Encryption in the Quantum Random Oracle Model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 758–775. Springer, 2012. URL: https://doi.org/10.1007/978-3-642-32009-5_44.

[Zha18] Mark Zhandry. How to Record Quantum Queries, and Applications to Quantum Indifferentiability. *IACR Cryptology ePrint Archive*, 2018:276, 2018. URL: https://eprint.iacr.org/2018/276.

Part II.

Publications

List of Publications

Refereed Conference Proceedings

[CDG+17a] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1825–1842. ACM, 2017. URL: https://doi.org/10.1145/3133956.3133997.

[DKL+18a] David Derler, Stephan Krenn, Thomas Lorünser, Sebastian Ramacher, Daniel Slamanig, and Christoph Striecks. Revisiting Proxy Re-encryption: Forward Secrecy, Improved Security, and Applications. In Michel Abdalla and Ricardo Dahab, editors, *Public-Key Cryptography - PKC 2018 - 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part I*, volume 10769 of *Lecture Notes in Computer Science*, pages 219–250. Springer, 2018. URL: https://doi.org/10.1007/978-3-319-76578-5_8.

[DKP+19] Itai Dinur, Daniel Kales, Angela Promitzer, Sebastian Ramacher, and Christian Rechberger. Linear Equivalence of Block Ciphers with Partial Non-Linear Layers: Application to LowMC. In *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, Lecture Notes in Computer Science, 2019. to appear.

[DRS17] David Derler, Sebastian Ramacher, and Daniel Slamanig. Homomorphic Proxy Re-Authenticators and Applications to Verifiable Multi-User Data Aggregation. In Aggelos Kiayias, editor, *Financial Cryptography and Data Security - 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers*, volume 10322 of *Lecture Notes in Computer Science*, pages 124–142. Springer, 2017. URL: https://doi.org/10.1007/978-3-319-70972-7_7.

[DRS18a] David Derler, Sebastian Ramacher, and Daniel Slamanig. Generic Double-Authentication Preventing Signatures and a Post-quantum Instantiation. In Joonsang Baek, Willy Susilo, and Jongkil Kim, editors, *Provable Security - 12th International Conference, ProvSec 2018, Jeju, South Korea, October 25-28, 2018, Proceedings*, volume 11192 of *Lecture Notes in Computer Science*, pages 258–276. Springer, 2018. URL: https://doi.org/10.1007/978-3-030-01446-9_15.

[DRS18b] David Derler, Sebastian Ramacher, and Daniel Slamanig. Post-Quantum Zero-Knowledge Proofs for Accumulators with Applications to Ring Signatures from Symmetric-Key Primitives. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings*, volume 10786 of *Lecture Notes in Computer Science*, pages 419–440. Springer, 2018. URL: https://doi.org/10.1007/978-3-319-79063-3_20.

[DRS18c] David Derler, Sebastian Ramacher, and Daniel Slamanig. Short Double- and N-Times-Authentication-Preventing Signatures from ECDSA and More. In *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018*, pages 273–287. IEEE, 2018. URL: https://doi.org/10.1109/EuroSP.2018.00027.

[KOR19] Daniel Kales, Olamide Omolola, and Sebastian Ramacher. Revisting User Privacy for Certificate Transparency. In *2019 IEEE European Symposium on Security and Privacy, EuroS&P 2019, Stockholm, Sweden, June 17-19, 2019.* IEEE, 2019. to appear.

Journal Publications

[GRR15] Alfred Geroldinger, Sebastian Ramacher, and Andreas Reinhart. On v-Marot Mori rings and C-rings. *Journal of the Korean Mathematical Society*, 52(1), 2015.

Preprints and Miscellaneous

[AGP$^+$19] Martin R. Albrecht, Lorenzo Grassi, Léo Perrin, Sebastian Ramacher, Christian Rechberger, Dragos Rotaru, Arnab Roy, and Markus Schofnegger. Feistel Structures for MPC, and More. *IACR Cryptology ePrint Archive*, 2019:397, 2019. URL: http://eprint.iacr.org/2019/397.

[CDG$^+$17b] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. The Picnic Signature Scheme: Design Document, 2017. URL: https://github.com/Microsoft/Picnic/blob/master/spec/design-v1.0.pdf.

[CDG$^+$19] Melissa Chase, David Derler, Steven Goldfeder, Jonathan Katz, Vladimir Kolesnikov, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Xiao Wang, and Greg Zaverucha. The Picnic Signature Scheme: Design Document (Version 2.0), 2019. URL: https://github.com/Microsoft/Picnic/blob/master/spec/design-v2.0.pdf.

[DOR$^+$16] David Derler, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, and Daniel Slamanig. Digital Signatures from Symmetric-Key Primitives. *IACR Cryptology ePrint Archive*, 2016:1085, 2016. URL: http://eprint.iacr.org/2016/1085.

[KPP$^+$17] Daniel Kales, Léo Perrin, Angela Promitzer, Sebastian Ramacher, and Christian Rechberger. Improvements to the Linear Layer of LowMC: A Faster Picnic. *IACR Cryptology ePrint Archive*, 2017:1148, 2017. URL: http://eprint.iacr.org/2017/1148.

3

Revisiting Proxy Re-Encryption: Forward Secrecy, Improved Security, and Applications

Publication Data

David Derler, Stephan Krenn, Thomas Lorünser, Sebastian Ramacher, Daniel Slamanig, and Christoph Striecks. Revisiting Proxy Re-encryption: Forward Secrecy, Improved Security, and Applications. In Michel Abdalla and Ricardo Dahab, editors, *Public-Key Cryptography - PKC 2018 - 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part I*, volume 10769 of *Lecture Notes in Computer Science*, pages 219–250. Springer, 2018. URL: https://doi.org/10.1007/978-3-319-76578-5_8

The appended paper is an author-created full version available at https://eprint.iacr.org/2018/321. The full version contains an additional construction of forward-secure proxy re-encryption from fully puncturable encryption.

Contributions

- The author is one of the main authors.

Revisiting Proxy Re-Encryption: Forward Secrecy, Improved Security, and Applications

David Derler[1], Stephan Krenn[2], Thomas Lorünser[2], Sebastian Ramacher[1], Daniel Slamanig[2], and Christoph Striecks[2]

[1] IAIK, Graz University of Technology, Austria
[2] AIT Austrian Institute of Technology, Vienna, Austria
{firstname.lastname}@tugraz.at, {firstname.lastname}@ait.ac.at

Abstract. We revisit the notion of proxy re-encryption (PRE), an enhanced public-key encryption primitive envisioned by Blaze et al. (EUROCRYPT'98) and formalized by Ateniese et al. (NDSS'05) for delegating decryption rights from a delegator to a delegatee using a semi-trusted proxy. PRE notably allows to craft re-encryption keys in order to equip the proxy with the power of transforming ciphertexts under a delegator's public key to ciphertexts under a delegatee's public key, while not learning anything about the underlying plaintexts.

We study an attractive cryptographic property for PRE, namely that of forward secrecy. In our forward-secret PRE (fs-PRE) definition, the proxy periodically evolves the re-encryption keys and permanently erases old versions while the delegator's public key is kept constant. As a consequence, ciphertexts for old periods are no longer re-encryptable and, in particular, cannot be decrypted anymore at the delegatee's end. Moreover, delegators evolve their secret keys too, and, thus, not even they can decrypt old ciphertexts once their key material from past periods has been deleted. This, as we will discuss, directly has application in short-term data/message-sharing scenarios.

Technically, we formalize fs-PRE. Thereby, we identify a subtle but significant gap in the well-established security model for conventional PRE and close it with our formalization (which we dub fs-PRE$^+$). We present the first provably secure and efficient constructions of fs-PRE as well as PRE (implied by the former) satisfying the strong fs-PRE$^+$ and PRE$^+$ notions, respectively. All our constructions are instantiable in the standard model under standard assumptions and our central building block are hierarchical identity-based encryption (HIBE) schemes that only need to be selectively secure.

Keywords: Forward secrecy, proxy re-encryption, improved security model

This is the full version of a paper which appears in Public-Key Cryptography - PKC 2018 - 21st IACR International Conference on Practice and Theory in Public-Key Cryptography, Rio De Janeiro, Brazil, March 25-28, 2018, Proceedings. ©IACR, 2018.

1 Introduction

The security of cryptosystems essentially relies on the secrecy of the respective secret key. For example, if for an encryption scheme a secret key is (accidentally) leaked, the confidentiality of all the data encrypted with respect to this key so far is immediately destroyed. One simple mitigation strategy for such a secret-key leakage is to frequently change secret keys such that leaking a secret key only affects a small amount of data. Implementing this in a naïve way, for instance in context of public-key encryption, means that one either has to securely and interactively distribute copies of new public keys frequently or to have huge public keys[3], which is rather inconvenient in practice. Consequently, cryptographic research focused on the design of cryptosystems that inherently provide such a property, being denoted as *forward secrecy* (or, *forward security*) [28]. The goal hereby is that key leakage at some point in time does not affect the data which was encrypted before the key leakage, while mitigating the drawbacks of the naïve solution discussed before. That is, one aims at efficient non-interactive solutions that have fixed sublinear-size public keys in the number of key switches/time periods. Those (strong) properties are the minimal requirements in the de-facto standard notion of forward secrecy in the cryptographic literature.

Within the last two decades, forward secrecy has been identified as an important property of various different cryptographic primitives such as digital signatures [6], identification schemes [1], public-key encryption [15], and private-key cryptography [7]. Only recently, another huge step forward has been made by Green and Miers [27] as well as Günther, Jager, Hale, and Lauer [29] to bring forward secrecy to important practical applications in the context of asynchronous messaging and zero round-trip time (0-RTT) key exchange. Given revelations and leaks about large-scale surveillance activities of security agencies within the last years, it is of utmost importance to further develop and deploy cryptosystems that inherently provide forward secrecy. We aim at advancing the research on forward secrecy with respect to other practically important public-key primitives, ideally, to ones with slightly more functionality.

Proxy re-encryption. Proxy re-encryption (PRE), envisoned by Blaze, Bleumer, and Strauss [9] and formalized by Ateniese, Fu, Green, and Hohenberger [4, 5], is a cryptographic primitive that can be seen as an extension of public-key encryption. A central feature of PRE is that senders can craft so-called re-encryption keys, which are usually created using only public information of the designated delegatee and the delegators' key material. Those re-encryption keys have the power to transform ciphertexts under a delegator's public key to ciphertexts under the delegatees' public keys. Within PRE, this transformation is done by a semi-trusted[4] proxy. The widely accepted model for PRE security (i.e.,

[3] With size $O(n)$ for n key switches/time periods.

[4] A semi-trusted proxy honestly follows the protocols, i.e., stores consistent re-encryption keys and re-encrypts correctly.

the conventional or plain PRE model) [4] requires that the proxy does not learn anything about the plaintexts which underlie the ciphertexts to be transformed.[5]

Proxy re-encryption is considered very useful in applications such as encrypted e-mail forwarding or access control in secure file systems, which was already discussed heavily in earlier work, e.g., in [4]. Furthermore, PRE has been object of significant research for almost two decades now, be it in a conventional setting [9, 4, 5], PRE with temporary delegation [4, 5, 34], identity-based PRE [26, 37], extensions to the chosen-ciphertext setting [16, 34], type-based/conditional PRE [39, 41], anonymous (or key-private) PRE [3], traceable PRE [32], or PRE from lattice-based assumptions [18, 36]. Generic constructions of PRE schemes from fully-homomorphic encryption [24] and from non-standard building blocks such as resplittable-threshold public key encryption as proposed in [30] are known, where different constructions of secure obfuscators for the re-encryption functionality have been given [31, 19, 18]. Despite PRE being an object of such significant research, forward-secret constructions remain unknown.[6]

On modeling forward-secret proxy re-encryption. Forward secrecy in the context of PRE is more complex than in standard public-key primitives, as PRE involves multiple different parties (i.e., delegator, proxy, and delegatees), where delegator and delegatees all have their own secret-key material and the proxy additionally holds all the re-encryption keys. One may observe that the proxy needs to be considered as a semi-trusted (central) party being always online, and, thus, it is reasonable to assume that this party is most valuable to attack. Consequently, we model forward secrecy in the sense that the re-encryption-key material can be evolved by the proxy to new periods while past-period re-encryption keys are securely erased. Hence, ciphertexts under the delegator's public key with respect to past-periods can no longer be re-encrypted. In addition, we model forward secrecy for the delegator's key material in a way that it is consistent with the evolution of the re-encryption material at the proxy.

For now, we do not consider forward secrecy at the delegatee, who can be seen as a passive party and does not need to take any further interaction with the delegator during the life-time of the system, except providing her public key once after set-up (e.g., via e-mail or public key server). It also does not have to be online when ciphertexts are re-encrypted for her by the proxy. Nevertheless, we leave it as a path for future research to cover the third dimension, i.e., model forward secrecy for the delegator and proxy as well as forward secrecy for the delegatee with efficient non-trivial constructions. However, it seems highly non-trivial to achieve efficient constructions that support forward secrecy for the delegatee additionally. In particular, we believe that the difficulty of achieving such strong type of forward secrecy is due to the circumstance that one has to

[5] The well-established security notions for PRE leave a potentially critical gap open. To look ahead, our proposed security model for *forward-secret* PRE closes this gap (implicitly also for plain PRE) and goes even beyond.

[6] We stress that we only aim at efficient non-trivial (non-interactive) forward-secret PRE constructions that have sublinear-size public and re-encryption keys in the number of time periods.

carefully integrate three dimension of evolving key-material, one at the delegator, one at the proxy, and one at the delegatee. All dimensions seem to interfere with each other.[7] As it will be confirmed by our application, covering the two dimensions already yields an interesting tool.

Moreover, to achieve forward secrecy for delegator and proxy key material, we face the following obstacles. First, it has to be guaranteed that the honest proxy must not be able to gain any information from the ciphertexts while at the same time being able to transform such ciphertexts *and* to update re-encryption key material consistently to newer time periods *without* any interaction with the delegator. Secondly, any delegatee *must not* be able to decrypt past-period ciphertexts. In this work, we give an affirmative answer to overcome those obstacles.

A practical application of forward-secret PRE. We believe that forward secrecy is an essential topic nowadays for any application. Also PRE is increasingly popular, be it in applied cryptographic literature [10, 14, 42, 36, 35], working groups such as the CFRG of the IRTF[8], large-scale EU-funded projects[9], and meanwhile also companies[10] that foster transition of such technologies into applications.

A practical application for forward-secret PRE is disappearing 1-to-n messaging. Here, a user encrypts a message under his public key and sends it to the proxy server that is responsible for distributing the encrypted messages to all pre-determined n receivers (note that receivers do not have to be online at the time the encrypted message is sent and an initial public-key exchange has to be done only in the beginning, but no more interactivity is needed). During setup time, the user has equipped the server with re-encryption keys (one for each receiver) while new keys can be added any time once a new receiver is present. Furthermore, the user does not need to manage a potentially huge list of public keys for each message to be sent. After a period, the data gets deleted by the proxy server, the re-encryption keys get evolved to a new period (without any interactions), and old-period re-encryption keys get deleted. The security of forward-secret PRE then guarantees that the proxy server does not learn the sensitive messages, neither can the two types of parties access disappeared messages later on. Once period-i re-encryption keys leak from the proxy server, only present and future encrypted messages (from period i onward) are compromised, while period-$(i-1)$ messages stay confidential. More generally, we believe that forward-secret PRE can be beneficially used in all kinds of settings that require access revocation, e.g., in outsourced encrypted data storage.

We also stress that within our forward-secret PRE instantiations, each user is only required to manage her own public and secret keys on her device and not a

[7] It is currently unknown to us how to solve the problem with efficient cryptographic tools, e.g., in the bilinear-maps setting. For efficiency reasons, multilinear maps and obfuscation are out of focus.

[8] https://www.ietf.org/id/draft-hallambaker-mesh-recrypt-00.txt

[9] https://credential.eu/

[10] e.g., http://www.nucypher.com, https://besafe.io/

list of recipient public keys (or, identities). This deviates significantly from other primitives such as broadcast encryption (BE) [12, 22, 38], which could also be suitable in such scenarios. However, practical BE schemes, e.g., [13], need large public keys and are computationally expensive.

1.1 Contribution

In this paper, we investigate forward secrecy in the field of proxy re-encryption (PRE) and term it fs-PRE. More precisely, our contributions are as follows:

- We first port the security model of PRE to the forward-secret setting (fs-PRE$^-$). Thereby, we observe a subtle but significant gap in existing (plain) security models for conventional PRE with regard to the granularity of delegations of decryption rights. In particular, existing models allow that a recipient, who has once decrypted a re-encrypted ciphertext, can potentially decrypt all re-encryptable ciphertexts of the same sender without further involvement of the proxy. In the forward-secret setting, it would essentially require to trust the delegatees to delete their re-encrypted ciphertexts whenever the period is switched, which is a problematic trust assumption.[11]
- We close this gap by introducing an additional security notion which inherently requires the involvement of a proxy in every re-encryption and in particular consider this notion in the forward-secret setting (fs-PRE$^+$). We also note that, when considering only a single time interval, this implicitly closes the aforementioned gap in the conventional PRE setting.[12] We also provide an explicit separation of the weaker fs-PRE$^-$ notion (resembling existing PRE models) and our stronger notion fs-PRE$^+$.
- We then continue by constructing the first forward-secret PRE schemes (in the weaker as well as our stronger model) that are secure in the standard model under standard assumptions. On a technical side, only few approaches to forward secrecy are known. Exemplary, in the public-key-encryption (PKE) setting, we essentially have two ways to construct forward secrecy, i.e., the Canetti-Halevi-Katz (CHK) framework [15] from selectively secure hierarchical identity-based encryption (HIBE) [25] schemes and the more abstract puncturable-encryption (PE) approaches by [27, 29] (where both works either explicitly or implicitly use the CHK techniques). Particularly, we are not aware of any framework to achieve forward secrecy for PKE schemes based on "less-complex" primitives in comparison to selectively secure HIBE schemes. Consequently, we also base our constructions on selectively secure HIBE schemes [25], which we combine with linearly homomorphic encryption schemes, e.g., (linear) ElGamal.
- As a side result, we generalize the recent work of PE [27, 21, 17, 29] to what we call fully puncturable encryption (FuPE) in the full version of this paper and show how we can use FuPE to construct fs-PRE.

[11] Clearly, we still have to trust that the proxy deletes past-period re-encryption key material.

[12] In the conventional PRE setting, this gap was very recently independently addressed by Cohen [20].

1.2 Intuition and Construction Overview

To obtain more general results and potentially also more efficient instantiations, we use a relaxation of HIBEs denoted as binary-tree encryption (BTE) which was introduced by Canetti, Halevi, and Katz (CHK) in [15]. As an intermediate step, we introduce the notion of a forward-secret delegatable public-key encryption (fs-DPKE) scheme and present one instantiation which we obtain by combining the results of CHK with a suitable homomorphic public-key encryption (HPKE) scheme. Loosely speaking, a fs-DPKE scheme allows to delegate the decryption functionality of ciphertexts computed with respect to the public key of some user A to the public key of some other user B. Therefore, A provides a *public* delegation key to B. B then uses the delegation key *together* with the secret key corresponding to B's public key to decrypt any ciphertext that has been produced for A. A fs-DPKE scheme moreover incorporates forward secrecy in a sense that the originator A can evolve it's secret key and the scheme additionally allows to *publicly* evolve delegation keys accordingly. Interestingly, such a scheme is already sufficient to construct a fs-PRE$^-$-secure PRE scheme. Finally, we demonstrate how to strengthen this construction to a fs-PRE$^+$-secure PRE scheme, by solely relying on a certain type of key-homomorphism of the underlying fs-DPKE scheme. The intermediate step of introducing fs-DPKE is straightforward yet interesting, since we believe fs-DPKE is the "next natural step" to lift PKE to a setting which allows for controlled delegation of decryption rights.

Instantiation. In Table 1, we present an instantiation including the resulting key and ciphertext sizes. Thereby, we only look at fs-PRE instantiations that are fs-PRE$^+$-secure and note that the asymptotic sizes for fs-PRE$^-$-secure fs-PRE schemes are identical. For our instantiation, we use the BTE (or any selectively secure HIBE) from [15] and the linear encryption scheme from [11] as HPKE scheme under the Bilinear Decisional Diffie-Hellman (BDDH) and decision linear (DLIN) assumption respectively.

Building Blocks	$\lvert \mathsf{pk} \rvert$	$\lvert \mathsf{rk}^{(i)} \rvert$	$\lvert \mathsf{sk}^{(i)} \rvert$	$\lvert C \rvert$	Assumption
BTE [15], HPKE [11]	$\mathcal{O}(\log n)$	$\mathcal{O}((\log n)^2)$	$\mathcal{O}((\log n)^2)$	$\mathcal{O}(\log n)$	BDDH, DLIN

Table 1. Our fs-PRE$^+$-secure instantiation. All parameters additionally scale asymptotically in a security parameter k which is, hence, omitted. Legend: n ... number of periods, $\lvert \mathsf{pk} \rvert$ ... public key size, $\lvert \mathsf{rk}^{(i)} \rvert$ ... size of re-encryption key for period i, $\lvert \mathsf{sk}^{(i)} \rvert$ ... size of secret key for period i, $\lvert C \rvert$ ... ciphertext size.

A note on a side result. Additionally, in the full version, we include the definition and a construction of a so called fully puncturable encryption (FuPE) scheme which is inspired by techniques known from HIBEs and the recent PE schemes in [27, 29]. We then show that FuPE schemes closely capture the essence which is required to construct fs-PRE$^+$-secure schemes by presenting a construction of a fs-PRE$^+$-secure PRE scheme from FuPE and HPKE.

1.3 Related Work and Outline

Work related to forward-secret PRE. Tang et al. [39, 41] introduced type-based/conditional PRE, which allows re-encryption of ciphertexts at the proxy only if a specific condition (e.g., a time period) is satisfied by the ciphertext. Furthermore, PRE with temporary delegations was proposed by Ateniese et al. [4, 5] and improved by Libert and Vernaud (LV) [34]. All those approaches yield a weak form of forward secrecy. Notably, the LV schemes provide fixed public parameters and non-interactivity with the delegatee as well. However, in contrast to our approach, LV and Tang et al. require at least to update the re-encryption keys for each time period with the help of the delegator (i.e., one message per time period from the delegator to the proxy) and also do not allow for exponentially many time periods, which do not suit our (stronger) forward-secret scenario.

Concurrent work on PRE. There is a considerable amount of very recent independent and concurrent work on different aspects of PRE and its applications [20, 8, 35, 23]. The works in [8, 35, 23] are only related in that they also deal with various aspects of PRE, but not fs-PRE. Those aspects are however unrelated to the work presented in this paper. In contrast, the work presented in [20] is related to one aspect of our work. It formalizes a security property for conventional PRE, which is stronger yet similar to a special case of the fs-PRE$^+$ notion which we introduce in context of fs-PRE.

Outline. After discussing preliminaries in Section 2, we define fs-PRE in Section 3, discuss the gap in previous models and also briefly discuss its consequences to conventional PRE. We then give the first construction of a fs-PRE scheme from binary tree encryption in Section 4. We also show a separation result for the weaker fs-PRE$^-$ (resembling existing PRE models) and our stronger notion fs-PRE$^+$.

2 Preliminaries

For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$ and let $k \in \mathbb{N}$ be the security parameter. For an algorithm A, let $y \leftarrow A(1^k, x)$ be the process of running A, on input 1^k and x, with access to uniformly random coins and assigning the result to y. We assume that all algorithms take 1^k as input and we will sometimes not make this explicit in the following. To make the random coins r explicit, we write $A(1^k, x; r)$. An algorithm A is probabilistic polynomial time (PPT) if its running time is polynomially bounded in k. A function f is negligible if $\forall c \exists k_0 \forall k \geq k_0 : |f(k)| \leq 1/k^c$. For binary trees, we denote the root node with ε and all other nodes are encoded as binary strings, i.e., for a node w we denote child nodes as $w0$ and $w1$.

Homomorphic public-key encryption. A $\mathcal{F}$-homomorphic public key encryption (HPKE) scheme is a public-key encryption (PKE) scheme that is homomorphic with respect to a class of functions $\mathcal{F}$, i.e., given a sequence of ciphertexts to messages $(M_i)_{i \in [n]}$ one can evaluate a function $f : \mathcal{M}^n \rightarrow \mathcal{M} \in \mathcal{F}$ on the ciphertexts such that the resulting ciphertext decrypts to $f(M_1, \ldots, M_n)$.

Definition 1 **(($\mathcal{F}$-)HPKE).** *A $\mathcal{F}$-homomorphic public key encryption ($\mathcal{F}$-HPKE or HPKE for short) scheme with message space $\mathcal{M}$, ciphertext space $\mathcal{C}$ and a function family $\mathcal{F}$ consists of the PPT algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$*:*

$\mathsf{Gen}(1^k)$: *On input security parameter k, outputs public and secret keys* $(\mathsf{pk}, \mathsf{sk})$*.*

$\mathsf{Enc}(\mathsf{pk}, M)$: *On input a public key pk, and a message $M \in \mathcal{M}$, outputs a ciphertext $C \in \mathcal{C}$.*

$\mathsf{Dec}(\mathsf{sk}, C)$: *On input a secret key sk, and ciphertext C, outputs $M \in \mathcal{M} \cup \{\perp\}$.*

$\mathsf{Eval}(f, (C_i)_{i\in[n]})$: *On input a function $f : \mathcal{M}^n \to \mathcal{M} \in \mathcal{F}$, a sequence of ciphertexts $(C_i)_{i\in[n]}$ encrypted under the same public key, outputs C.*

In addition to the standard and folklore correctness definition for public-key encryption (PKE), we further require for HPKE that for all security parameters $k \in \mathbb{N}$, all key pairs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^k)$, all functions $f : \mathcal{M}^n \to \mathcal{M} \in \mathcal{F}$, all message sequences $(M_i)_{i\in[n]}$ it holds that $\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(f, (\mathsf{Enc}(\mathsf{pk}, M_i))_{i\in[n]})) = f(M_1, \ldots, M_n)$. We are particularly interested in the case where $\mathcal{M}$ is a group and $\mathcal{F}$ is the set of all *linear functions* on products of $\mathcal{M}$. In that case, we call the HPKE scheme *linearly homomorphic*. For a HPKE, we require conventional IND-CPA security as with PKE schemes and recall an efficient instantiation of a linearly homomorphic scheme, i.e., linear ElGamal [11], in the full version.

Proxy re-encryption. Subsequently, we define proxy re-encryption.

Definition 2 **(PRE).** *A proxy re-encryption (PRE) scheme with message space $\mathcal{M}$ consists of the PPT algorithms* $(\mathsf{Setup}, \mathsf{Gen}, \mathbf{Enc}, \mathbf{Dec}, \mathsf{ReGen}, \mathsf{ReEnc})$ *where* $\mathbf{Enc} = (\mathsf{Enc}^{(j)})_{j\in[2]}$ *and* $\mathbf{Dec} = (\mathsf{Dec}^{(j)})_{j\in[2]}$*. For $j \in [2]$, they are defined as follows.*

$\mathsf{Setup}(1^k)$: *On input security parameter k, outputs public parameters* pp*.*

$\mathsf{Gen}(\mathsf{pp})$: *On input public parameters pp, outputs public and secret keys* $(\mathsf{pk}, \mathsf{sk})$*.*

$\mathsf{Enc}^{(j)}(\mathsf{pk}, M)$: *On input a public key pk, and a message $M \in \mathcal{M}$ outputs a level j ciphertext C.*

$\mathsf{Dec}^{(j)}(\mathsf{sk}, C)$: *On input a secret key sk, and level j ciphertext C, outputs $M \in \mathcal{M} \cup \{\perp\}$.*

$\mathsf{ReGen}(\mathsf{sk}_A, \mathsf{pk}_B)$: *On input a secret key sk_A and a public key pk_B for B, outputs a re-encryption $\mathsf{rk}_{A\to B}$.*

$\mathsf{ReEnc}(\mathsf{rk}_{A\to B}, C_A)$: *On input a re-encryption key $\mathsf{rk}_{A\to B}$, and a ciphertext C_A for user A, outputs a ciphertext C_B for user B.*

Subsequently, we restate the standard security notions of proxy re-encryption schemes [4, 5, 33]. The oracles available in the experiment are as follows. For all experiments defined in this section, the environment keeps initially empty lists of dishonest ($\mathtt{DU}$) and honest users ($\mathtt{HU}$). The oracles are defined as follows:

$\mathsf{Gen}^{(h)}(\mathsf{pp}, n)$: Run $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n)$, set $\mathtt{HU} \leftarrow \mathtt{HU} \cup \{(\mathsf{pk}, \mathsf{sk})\}$, and return pk.

$\mathsf{Gen}^{(d)}(\mathsf{pp}, n)$: Run $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n)$, set $\mathtt{DU} \leftarrow \mathtt{DU} \cup \{(\mathsf{pk}, \mathsf{sk})\}$, and return $(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{ReGen}^{(h)}(\mathsf{pk}_u, \mathsf{pk})$: On input a public key pk_u and a public key pk, abort if $(\mathsf{pk}_u, \cdot) \notin \mathtt{HU}$. Otherwise, look up sk_u corresponding to pk_u from $\mathtt{HU}$. Return $\mathsf{ReGen}(\mathsf{sk}_u, \mathsf{pk})$.

$\mathsf{ReGen}^{(h')}(\mathsf{sk}, \mathsf{pk}_u)$: On input a secret key sk and a public key pk_u, abort if $(\mathsf{pk}_u, \cdot) \notin \mathtt{HU}$. Otherwise, return $\mathsf{ReGen}(\mathsf{sk}, \mathsf{pk}_u)$.

$\mathsf{ReGen}^{(d)}(\mathsf{sk}, \mathsf{pk}_d)$: On input a secret key sk and a public key pk_d, abort if $(\mathsf{pk}_d, \cdot) \notin \mathtt{DU}$. Otherwise, return $\mathsf{ReGen}(\mathsf{sk}, \mathsf{pk}_d)$.

```
Experiment $\mathsf{Exp}^{\mathsf{ind\text{-}cpa\text{-}1}}_{\mathsf{PRE},A}(1^k)$
  $\mathsf{pp} \leftarrow \mathsf{Setup}(1^k), (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{pp}), b \xleftarrow{R} \{0,1\}$
  $\mathcal{O} \leftarrow \{\mathsf{Gen}^h, \mathsf{ReGen}^h(\cdot, \mathsf{pk}), \mathsf{ReGen}^{h'}(\mathsf{sk}, \cdot), \mathsf{Gen}^d, \mathsf{ReGen}^d(\mathsf{sk}, \cdot)\}$
  $(M_0, M_1, \mathsf{st}) \leftarrow A^{\mathcal{O}}(\mathsf{pk})$
  $b^* \leftarrow A(\mathsf{st}, \mathsf{Enc}^{(1)}(\mathsf{pk}, M_b))$
  if $b = b^*$ return 1, else return 0
```

Experiment 1. The IND-CPA security experiment for level 1 ciphertexts of fs-PRE schemes.

Definition 3 (IND-CPA-1). *For a PPT adversary A, we define the advantage function in the sense of IND-CPA for level 1 ciphertexts as*

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa\text{-}1}}_{\mathsf{PRE},A}(1^k) := \left| \Pr\left[\mathsf{Exp}^{\mathsf{ind\text{-}cpa\text{-}1}}_{\mathsf{PRE},A}(1^k) = 1 \right] - \frac{1}{2} \right| .$$

If for any PPT adversary A there exists a negligible function ε such that

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa\text{-}1}}_{\mathsf{PRE},A}(1^k) < \varepsilon(k)$$

then a PRE *scheme is IND-CPA-1 secure.*

```
Experiment $\mathsf{Exp}^{\mathsf{ind\text{-}cpa\text{-}2}}_{\mathsf{PRE},A}(1^k)$
  $\mathsf{pp} \leftarrow \mathsf{Setup}(1^k)$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{pp})$, $b \xleftarrow{R} \{0,1\}$
  $\mathcal{O} \leftarrow \{\mathsf{Gen}^h, \mathsf{ReGen}^h(\cdot, \mathsf{pk}), \mathsf{ReGen}^{h'}(\mathsf{sk}, \cdot)\}$
  $(M_0, M_1, \mathsf{st}) \leftarrow A^{\mathcal{O}}(\mathsf{pk})$
  $b^* \leftarrow A(\mathsf{st}, \mathsf{Enc}^{(2)}(\mathsf{pk}, M_b))$
  if $b = b^*$ return 1, else return 0
```

Experiment 2. The IND-CPA security experiment for level 2 ciphertexts of PRE schemes.

Definition 4 (IND-CPA-2). *For a polynomially bounded function n, a PPT adversary A, we define the advantage function in the sense of IND-CPA for level 2 ciphertexts as*

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa\text{-}2}}_{\mathsf{PRE},A}(1^k) := \left| \Pr\left[\mathsf{Exp}^{\mathsf{ind\text{-}cpa\text{-}2}}_{\mathsf{PRE},2,A}(1^k) = 1 \right] - \frac{1}{2} \right| .$$

If for all polynomially bounded functions n, and any PPT adversary A there exists a negligible function ε such that

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa\text{-}2}}_{\mathsf{PRE},A}(1^k) < \varepsilon(k)$$

then a PRE *scheme is IND-CPA-2 secure.*

Binary tree encryption. Binary tree encryption (BTE) [15] is a relaxed version of hierarchical identity-based encryption (HIBE) [25]. Similar to a HIBE scheme, a BTE scheme has a (master) public key associated to a binary tree where each node in the tree has a corresponding secret key. To encrypt a message for some node, one uses both the public key and the name of the target node. Using the node's secret key, the resulting ciphertext can then be decrypted. Additionally, the secret key of a node can be used to derive the secret keys of its child nodes.

In contrast to BTE defined in [15], we make the part of the secret key used to perform the key derivation explicit, i.e., we will have secret keys for the decryption and derivation keys to derive secret keys. In case, an instantiation does not support a clear distinction, it is always possible to assume that the derivation key is empty and everything is contained in the secret key.

Definition 5. *A binary tree encryption (*BTE*) scheme with message space* $\mathcal{M}$ *consists of the PPT algorithms* $(\mathsf{Gen}, \mathsf{Evo}, \mathsf{Enc}, \mathsf{Dec})$ *as follows:*

$\mathsf{Gen}(1^k, \ell)$: *On input security parameter* k *and depth of the tree* ℓ*, outputs public, secret, and derivation keys* $(\mathsf{pk}, \mathsf{sk}^{(\varepsilon)}, \mathsf{dk}^{(\varepsilon)})$*.*

$\mathsf{Der}(\mathsf{sk}^{(w)}, \mathsf{dk}^{(w)})$: *On input secret key* $\mathsf{sk}^{(w)}$ *and derivation key* $\mathsf{dk}^{(w)}$*, for node* $w \in \{0,1\}^{<\ell}$*, outputs secret keys* $\mathsf{sk}^{(w0)}, \mathsf{sk}^{(w1)}$ *and derivation keys* $\mathsf{dk}^{(w0)}$*,* $\mathsf{dk}^{(w1)}$ *for the two children of* w*.*

$\mathsf{Enc}(\mathsf{pk}, M, w)$: *On input a public key* pk*, a message* $M \in \mathcal{M}$*, and node* $w \in \{0,1\}^{\leq \ell}$*, outputs a ciphertext* C*.*

$\mathsf{Dec}(\mathsf{sk}^{(w)}, C)$: *On input a secret key* $\mathsf{sk}^{(w)}$*, for node* $w \in \{0,1\}^{\leq \ell}$*, and ciphertext* C*, outputs* $M \in \mathcal{M} \cup \{\perp\}$*.*

For correctness, we require that for all security parameters $k \in \mathbb{N}$, all depths $\ell \in \mathbb{N}$, all key pairs $(\mathsf{pk}, (\mathsf{sk}^{(\varepsilon)}, \mathsf{ek}^{(\varepsilon)}))$ generated by $\mathsf{Gen}(1^k, \ell)$, any node $w \in \{0,1\}^{\leq \ell}$, any derived key $\mathsf{sk}^{(w)}$ derived using Der from $(\mathsf{sk}^{(\varepsilon)}, \mathsf{dk}^{(\varepsilon)})$, and all messages $M \in \mathcal{M}$, it holds that $\mathsf{Dec}(\mathsf{sk}^{(w)}, \mathsf{Enc}(\mathsf{pk}, M, w)) = M$.

The indistinguishability against selective node, chosen plaintext attacks (IND-SN-CPA) is a generalization of the standard IND-CPA security notion of PKE schemes. Essentially, the security notion requires the adversary to commit to the node to be attacked in advance. The adversary gets access to all secret keys except the secret keys for all nodes that are on the path from the root node to the targeted node.

Definition 6 (IND-SN-CPA). *For a polynomially bounded function* ℓ*, a PPT adversary* A*, we define the advantage function in the sense of IND-SN-CPA as*

$$\mathsf{Adv}^{\mathsf{ind\text{-}sn\text{-}cpa}}_{\mathsf{BTE},A}(1^k, \ell(k)) = \left| \Pr\left[\mathsf{Exp}^{\mathsf{ind\text{-}sn\text{-}cpa}}_{\mathsf{BTE},A}(1^k, \ell(k)) = 1 \right] - \frac{1}{2} \right|.$$

If for all ℓ*, and any* A *there exists a negligible function* ε *such that* $\mathsf{Adv}^{\mathsf{ind\text{-}sn\text{-}cpa}}_{\mathsf{BTE},A}($ $1^k, \ell(k)) < \varepsilon(k)$*, then a* BTE *scheme is IND-SN-CPA secure.*

```
Experiment $\mathsf{Exp}^{\mathsf{ind\text{-}sn\text{-}cpa}}_{\mathsf{BTE},A}(1^k, \ell)$
$(\mathsf{pk}, \mathsf{sk}^{(\varepsilon)}, \mathsf{dk}^{(\varepsilon)}) \leftarrow \mathsf{Gen}(1^k, \ell)$
$b \xleftarrow{R} \{0, 1\}$
$(w^*, \mathsf{st}) \leftarrow A(1^k, \ell)$
Let $W$ be the set of all nodes that are siblings to the path from the root node to $w^*$
and (if possible) $w^*0$ and $w^*1$.
Compute $(\mathsf{sk}^{(w)}, \mathsf{dk}^{(w)})$ for all $w \in W$ from $(\mathsf{sk}^{(\varepsilon)}, \mathsf{dk}^{(\varepsilon)})$ using $\mathsf{Der}$.
$(M_0, M_1, \mathsf{st}) \leftarrow A(\mathsf{st}, \mathsf{pk}, (\mathsf{sk}^{(w)}, \mathsf{dk}^{(w)})_{w \in W})$
$b^* \leftarrow A(\mathsf{st}, \mathsf{Enc}(\mathsf{pk}, M_b, w^*))$
if $b = b^*$ return 1, else return 0
```

Experiment 3. The IND-SN-CPA security experiment for a BTE scheme.

The CHK Compiler. The technique of Canetti et al. [15] can be summarized as follows. To build a forward-secret PKE scheme with n periods, one uses a BTE of depth ℓ such that $n < 2^{\ell+1}$. Associate each period with a node of the tree and write w^i to denote the node for period i. The node for period 0 is the root node, i.e. $w^0 = \varepsilon$. If w^i is an internal node, then set $w^{i+1} = w^i0$. Otherwise, if w^i is a leaf node and $i < N - 1$, then set $w^{i+1} = w'1$ where w' is the longest string such that $w'0$ is a prefix of w^i. The public key is simply the public key of the BTE scheme. The secret key for period i consists of the secret key for node w^i.

3 Security of (Forward-Secret) Proxy Re-Encryption

Proxy re-encryption (PRE) schemes can exhibit several important properties. In the following, we focus on the most common PRE properties in the cryptographic literature, i.e., uni-directionality (Alice is able to delegate decryption rights to Bob but not from Bob to Alice), non-interactivity (Alice can generate delegation key material without interacting with Bob), and collusion-safeness (even if Bob and other delegatees are colluding with the proxy, they cannot extract Alice' full secret key). Moreover, we consider PRE schemes that only allow a single hop, i.e., a ciphertext can be re-encrypted only a single time in contrast to multiple times in a row (multi-hop). Latter can be problematic due to unwanted transitivity.

In this work, we examine a further property of PRE schemes, namely the property of forward secrecy and propose the first uni-directional, non-interactive, collusion-safe, single hop, and forward-secret PRE scheme (dubbed fs-PRE) in the standard model from generic assumptions. Subsequently, in Section 3.1, we present the formal model for fs-PRE, while in Section 3.3 we discuss the relation and application of our stronger model to the conventional (i.e., plain) PRE security model.

3.1 Syntax of Forward-Secret Proxy Re-Encryption

To realize forward-secure PRE (fs-PRE), we lift the definitions and security models of uni-directional, single-hop, non-interactive, and collusion-safe PRE to a setting where we can have several periods. Thereby, we allow re-encryptions in every period such that re-encryption keys—in the same way as secret keys—are bound to a period. Furthermore, we align our PRE definitions with Ateniese et al. as well as Libert and Vergnaud [4, 5, 33] such that if we only have a single period, then they are equivalent to the definitions for plain PRE in [5, 33].[13]

Definition 7 (fs-PRE). *A forward-secure proxy re-encryption (fs-PRE) scheme with message space $\mathcal{M}$ consists of the PPT algorithms* $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Evo}, \mathbf{Enc}, \mathbf{Dec}, \mathsf{ReGen}, \mathsf{ReEvo}, \mathsf{ReEnc})$ *where* $\mathbf{Enc} = (\mathsf{Enc}^{(j)})_{j \in [2]}$ *and* $\mathbf{Dec} = (\mathsf{Dec}^{(j)})_{j \in [2]}$ *for levels $j \in [2]$. We denote level-2 ciphertext as re-encryptable ciphertexts, whereas level-1 ciphertexts are not re-encryptable.*

$\mathsf{Setup}(1^k)$: *On input security parameter k, outputs public parameters* pp.

$\mathsf{Gen}(\mathsf{pp}, n)$: *On input public parameters* pp*, and number of periods $n \in \mathbb{N}$, outputs public and secret keys* $(\mathsf{pk}, (\mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}))$.

$\mathsf{Evo}(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})$: *On input secret key* $\mathsf{sk}^{(i)}$ *and evolution key* $\mathsf{ek}^{(i)}$ *for period $i \in \{0, \ldots, n-2\}$, outputs a secret key* $\mathsf{sk}^{(i+1)}$ *and evolution key* $\mathsf{ek}^{(i+1)}$ *for period $i + 1$.*

$\mathsf{Enc}^{(j)}(\mathsf{pk}, M, i)$: *On input a public key* pk*, a message $M \in \mathcal{M}$, and period $i \in \{0, \ldots, n-1\}$, outputs a level-j ciphertext C.*

$\mathsf{Dec}^{(j)}(\mathsf{sk}^{(i)}, C)$: *On input a secret key* $\mathsf{sk}^{(i)}$*, for period $i \in \{0, \ldots, n-1\}$, and level-j ciphertext C, outputs $M \in \mathcal{M} \cup \{\perp\}$.*

$\mathsf{ReGen}(\mathsf{sk}_A^{(i)}, \mathsf{ek}_A^{(i)}, \mathsf{pk}_B)$: *On input a secret key* $\mathsf{sk}_A^{(i)}$ *and a evolution key* $\mathsf{ek}_A^{(i)}$ *(or $\perp$) for A and period $i \in \{0, \ldots, n-1\}$, and a public key* pk_B *for B, outputs a re-encryption* $\mathsf{rk}_{A \to B}^{(i)}$ *and re-encryption-evolution key* $\mathsf{rek}_{A \to B}^{(i)}$ *(or $\perp$).*

$\mathsf{ReEvo}(\mathsf{rk}_{A \to B}^{(i)}, \mathsf{rek}_{A \to B}^{(i)})$: *On input a re-encryption key* $\mathsf{rk}_{A \to B}^{(i)}$*, and a re-encryption-evolution key* $\mathsf{rek}_{A \to B}^{(i)}$ *for period $i \in \{0, \ldots, n-2\}$, outputs a re-encryption key* $\mathsf{rk}_{A \to B}^{(i+1)}$ *and re-encryption evolution key* $\mathsf{rek}_{A \to B}^{(i+1)}$ *for the period $i + 1$.*

$\mathsf{ReEnc}(\mathsf{rk}_{A \to B}^{(i)}, C_A)$: *On input a re-encryption key* $\mathsf{rk}_{A \to B}^{(i)}$*, and a (level-2) ciphertext C_A for user A, outputs a (level-1) ciphertext C_B for user B.*

Correctness. For correctness, we basically require on the one hand that every ciphertext encrypted for some period i can be decrypted with the respective secret key from period i. On the other hand—when also considering re-encryptable and re-encrypted ciphertexts—we require that level-2 ciphertexts encrypted for period i can be re-encrypted with a suitable re-encryption key for the same period and then decrypted using the (delegatee's) respective secret key for period i. More formally, for all security parameters $k \in \mathbb{N}$, all public parameters $\mathsf{pp} \leftarrow \mathsf{Setup}(1^k)$, any number of periods $n \in \mathbb{N}$ and users $U \in \mathbb{N}$, all key tuples $(\mathsf{pk}_u, \mathsf{sk}_u^{(0)}, \mathsf{ek}_u^{(0)})_{u \in [U]}$ generated by $\mathsf{Gen}(1^k, n)$, any period $i \in \{0, \ldots, n-1\}$, for any $u \in [U]$, any evolved key $\mathsf{sk}_u^{(i+1)}$ generated by $\mathsf{Evo}(\mathsf{sk}_u^{(i)})$, for all $u' \in [U], u \neq u'$, any (potentially evolved) re-encryption and

[13] Observe that for a single period, i.e., $n = 1$, Evo and ReEvo in Definition 7 are not defined. Dropping these algorithms and the corresponding evolution keys ek and rek yields a plain PRE scheme.

re-encryption-evolution keys $\mathsf{rk}_{u\to u'}^{(i)}$ and $\mathsf{rek}_{u\to u'}^{(i)}$, respectively, for period i generated using ReGen from (potentially evolved) secret and evolution keys as well as the target public key, and all messages $M \in \mathcal{M}$, it holds that

$$\forall j \in [2]\ \exists j' \in [2]:\ \mathsf{Dec}^{(j')}(\mathsf{sk}_u^{(i)}, \mathsf{Enc}^{(j)}(\mathsf{pk}_u, M, i)) = M,$$
$$\mathsf{Dec}^{(1)}(\mathsf{sk}_{u'}^{(i)}, \mathsf{ReEnc}(\mathsf{rk}_{u\to u'}^{(i)}, \mathsf{Enc}^{(2)}(\mathsf{pk}_u, M, i))) = M.$$

3.2 Security of Forward-Secret Proxy Re-Encryption

The security notions for fs-PRE are heavily inspired by the security notions of (plain) PRE [4, 5, 33] and forward-secret PKE [15]. We will discuss multiple notions, combine them carefully, and introduce forward-secret indistinguishably under chosen-plaintext attacks for level-1 and level-2 ciphertexts (termed fs-IND-CPA-1 and fs-IND-CPA-2, respectively) which we argue to be reasonable notions in our setting. Additionally, we define a new (stronger) variant of indistinguishably-under-chosen-plaintext-attacks security for fs-PRE (dubbed fs-RIND-CPA) that focuses on malicious users in the face of honest proxies. In particular, the latter strengthen the folklore PRE security notion.

For all experiments defined in this section, the environment keeps initially empty lists of dishonest (DU) and honest users (HU). The oracles are defined as follows:

$\mathsf{Gen}^{(h)}(\mathsf{pp}, n)$: Run $(\mathsf{pk}, \mathsf{sk}, \mathsf{ek}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n)$, set $\mathtt{HU} \leftarrow \mathtt{HU} \cup \{(\mathsf{pk}, \mathsf{sk}, \mathsf{ek})\}$, and return pk.

$\mathsf{Gen}^{(d)}(\mathsf{pp}, n)$: Run $(\mathsf{pk}, \mathsf{sk}, \mathsf{ek}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n)$, set $\mathtt{DU} \leftarrow \mathtt{DU} \cup \{(\mathsf{pk}, \mathsf{sk}, \mathsf{ek})\}$, and return $(\mathsf{pk}, \mathsf{sk}, \mathsf{ek})$.

$\mathsf{ReGen}^{(h)}(j, \mathsf{pk}_u, \mathsf{pk})$: On input a period j, a public key pk_u and a public key pk, abort if $(\mathsf{pk}_u, \cdot, \cdot) \notin \mathtt{HU}$. Otherwise, look up $\mathsf{sk}_u^{(0)}$ and $\mathsf{ek}_u^{(0)}$ corresponding to pk_u from $\mathtt{HU}$. If $j > 0$ set $(\mathsf{sk}_u^{(i)}, \mathsf{ek}_u^{(i)}) \leftarrow \mathsf{Evo}(\mathsf{sk}_u^{(i-1)}, \mathsf{ek}_u^{(i-1)})$ for $i \in [j]$. Return $\mathsf{ReGen}(\mathsf{sk}_u^{(j)}, \mathsf{ek}_u^{(j)}, \mathsf{pk})$.

$\mathsf{ReGen}^{(h')}(j, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \mathsf{pk}_u)$: On input a period j, secret key $\mathsf{sk}^{(0)}$, evolution key $\mathsf{ek}^{(0)}$, and a public key pk_u, abort if $(\mathsf{pk}_u, \cdot, \cdot) \notin \mathtt{HU}$. Otherwise, if $j > 0$ set $(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)}) \leftarrow \mathsf{Evo}(\mathsf{sk}^{(i-1)}, \mathsf{ek}^{(i-1)})$ for $i \in [j]$. Return $\mathsf{ReGen}(\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)}, \mathsf{pk}_u)$.

$\mathsf{ReGen}^{(d)}(j, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \mathsf{pk}_d)$: On input a period j, secret key $\mathsf{sk}^{(0)}$, evolution key $\mathsf{ek}^{(0)}$, and a public key pk_d, abort if $(\mathsf{pk}_d, \cdot, \cdot) \notin \mathtt{DU}$. Otherwise, if $j > 0$ set $(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)}) \leftarrow \mathsf{Evo}(\mathsf{sk}^{(i-1)}, \mathsf{ek}^{(i-1)})$ for $i \in [j]$. Return $\mathsf{ReGen}(\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)}, \mathsf{pk}_d)$.

fs-IND-CPA-i security. We start with the definition of fs-IND-CPA-1 and fs-IND-CPA-2 security for fs-PRE. Inspired by the work on forward secrecy due to Canetti, Halevi, and Katz [15], our experiments lift standard PRE security notions as defined in Ateniese et al. [4] (AFGH) to the forward-secrecy setting. More concretely, after the selection of a target period j^* by the adversary A, A gets access to the secret and the evolution key $(\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$ of the target period j^*, while the challenge ciphertext for A-chosen message M_b is generated for period $j^* - 1$, for uniform $b \leftarrow \{0,1\}$. Eventually, A outputs a guess on b. We say A is valid if A only outputs equal-length messages $|M_0| = |M_1|$ and $1 \leq j^* \leq n$.

Furthermore, we adapted the AFGH security experiment such that A has access to re-encryption and re-encryption-evolution keys for period $j^* - 1$. Analogously to previous work on PRE, we present two separate notions for level-1 and level-2 ciphertexts. The corresponding security experiments are given in Experiment 4 and Experiment 5. The only difference in Experiment 4 is that for level-1 ciphertexts, i.e., the ones which can no longer be re-encrypted, the adversary gets access to more re-encryption and re-encryption-evolution keys (obviously, the challenge ciphertext in that experiment is a level-1 ciphertext).

```
**Experiment** $\mathsf{Exp}^{\text{fs-ind-cpa-1}}_{\text{fs-PRE},A}(1^k, n)$
  $\mathsf{pp} \leftarrow \mathsf{Setup}(1^k), (\mathsf{pk}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n), b \xleftarrow{R} \{0,1\}$
  $(j^*, \mathsf{st}) \leftarrow A(\mathsf{pp}, n, \mathsf{pk})$
  $(\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)}) \leftarrow \mathsf{Evo}(\mathsf{sk}^{(j-1)}, \mathsf{ek}^{(j-1)})$ for $j \in [j^*]$.
  $\mathcal{O} \leftarrow \{\mathsf{Gen}^{(h)}, \mathsf{ReGen}^{(h)}(j^* - 1, \cdot, \mathsf{pk}), \mathsf{ReGen}^{(h')}(j^* - 1, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \cdot), \mathsf{Gen}^{(d)}, \mathsf{ReGen}^{(d)}(j^* - 1, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \cdot)\}$
  $(M_0, M_1, \mathsf{st}) \leftarrow A^{\mathcal{O}}(\mathsf{st}, \mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$
  $b^* \leftarrow A(\mathsf{st}, \mathsf{Enc}^{(1)}(\mathsf{pk}, M_b, j^* - 1))$
  if $b = b^*$ return 1, else return 0
```

Experiment 4. The fs-IND-CPA-1 security experiment for level-1 ciphertexts of fs-PRE schemes.

```
**Experiment** $\mathsf{Exp}^{\text{fs-ind-cpa-2}}_{\text{fs-PRE},A}(1^k, n)$
  $\mathsf{pp} \leftarrow \mathsf{Setup}(1^k), (\mathsf{pk}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n), b \xleftarrow{R} \{0,1\}$
  $(j^*, \mathsf{st}) \leftarrow A(\mathsf{pp}, n, \mathsf{pk})$
  $(\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)}) \leftarrow \mathsf{Evo}(\mathsf{sk}^{(j-1)}, \mathsf{ek}^{(j-1)})$ for $j \in [j^*]$.
  $\mathcal{O} \leftarrow \{\mathsf{Gen}^{(h)}, \mathsf{ReGen}^{(h)}(j^* - 1, \cdot, \mathsf{pk}), \mathsf{ReGen}^{(h')}(j^* - 1, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \cdot)\}$
  $(M_0, M_1, \mathsf{st}) \leftarrow A^{\mathcal{O}}(\mathsf{st}, \mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$
  $b^* \leftarrow A(\mathsf{st}, \mathsf{Enc}^{(2)}(\mathsf{pk}, M_b, j^* - 1))$
  if $b = b^*$ return 1, else return 0
```

Experiment 5. The fs-IND-CPA-2 security experiment for level-2 ciphertexts of fs-PRE schemes.

Definition 8 (fs-IND-CPA-i). *For a polynomially bounded function $n(\cdot) > 1$, a PPT adversary A, we define the advantage function for A in the sense of fs-IND-CPA-i for level-i ciphertexts as*

$$\mathsf{Adv}^{\text{fs-ind-cpa-i}}_{fs-\text{PRE},A}(1^k, n(k)) := \left| \Pr\left[\mathsf{Exp}^{\text{fs-ind-cpa}-\text{i}}_{fs-\text{PRE},A}(1^k, n(k)) = 1 \right] - \frac{1}{2} \right| .$$

*A fs-*PRE *scheme is fs-IND-CPA-i secure if for all polynomially bounded $n(\cdot) > 1$ and any valid PPT A there exists a negligible function ε such that $\mathsf{Adv}^{\text{fs-ind-cpa-i}}_{fs-\text{PRE},A}($ $1^k, n(k)) < \varepsilon(k)$, where $\mathsf{Exp}^{\text{fs-ind-cpa}-\text{i}}_{fs-\text{PRE},A}$, for all $i \in [2]$, are defined in Experiment 4 and Experiment 5, respectively.*

Master-secret security. As discussed in [33], the security notion for level-1 (i.e., non re-encryptable) ciphertexts already implies classical master-secret security notion for PRE [4].[14] However, this must not be the case in the forward-secret setting. To formally close this gap, we give a trivial lemma (cf. Lemma 1) which states that fs-IND-CPA-1 implies master-secret security in the sense of Experiment 6 in the forward-secrecy setting. Essentially, master-secret security ensures collusion safeness such that re-encryption keys in period j do not leak the secret key corresponding to level-1 ciphertexts which can not be re-encrypted in period $j-1$. In Experiment 6, we lift master-secret security in the classical PRE sense to the forward-secret setting. In the experiment, the adversary A selects an target period j^* and receives the secret and evolution keys $(\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$ for the target period in return. Within the experiment, A has access to several oracles, e.g., to obtain re-encryption and re-encryption-evolution keys for period j^*. Eventually, A outputs secret and evolutions keys $(\mathsf{sk}^*, \mathsf{ek}^*)$ and the experiment returns 1 (i.e., A wins) if $(\mathsf{sk}^*, \mathsf{ek}^*) = (\mathsf{sk}^{(j^*-1)}, \mathsf{ek}^{(j^*-1)})$. We say A is valid if A only outputs $1 \leq j^* \leq n$.

```
**Experiment** $\mathsf{Exp}^{\text{fs-msk}}_{fs-\mathsf{PRE},A}(1^k, n)$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^k), (\mathsf{pk}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n)$
$(j^*, \mathsf{st}) \leftarrow A(\mathsf{pp}, n, \mathsf{pk})$
$(\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)}) \leftarrow \mathsf{Evo}(\mathsf{sk}^{(j-1)}, \mathsf{ek}^{(j-1)})$ for $j \in [j^*]$.
$\mathcal{O} \leftarrow \{\mathsf{Gen}^{(h)}, \mathsf{ReGen}^{(h)}(j^*, \cdot, \mathsf{pk}), \mathsf{ReGen}^{(h')}(j^*, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \cdot), \mathsf{Gen}^{(d)}, \mathsf{ReGen}^{(d)}(j^*, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \cdot)\}$
$(\mathsf{sk}^*, \mathsf{ek}^*) \leftarrow A^{\mathcal{O}}(\mathsf{st}, \mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$
if $(\mathsf{sk}^*, \mathsf{ek}^*) = (\mathsf{sk}^{(j^*-1)}, \mathsf{ek}^{(j^*-1)})$ return 1, else return 0
```

Experiment 6. The forward secure master secret security experiment for fs-PRE schemes.

Definition 9 (fs-master-secret security). *For a polynomially bounded function $n(\cdot) > 1$ and a PPT adversary A, we define the advantage function for A in the sense of fs-master-secret security as*

$$\mathsf{Adv}^{\text{fs-msk}}_{fs-\mathsf{PRE},A}(1^k, n(k)) := \Pr\left[\mathsf{Exp}^{\text{fs-msk}}_{fs-\mathsf{PRE},A}(1^k, n(k)) = 1\right].$$

A fs-PRE scheme is fs-master-secret secure if for all polynomially bounded $n(\cdot) > 1$ and any valid PPT A there exists a negligible function ε such that $\mathsf{Adv}^{\text{fs-msk}}_{fs-\mathsf{PRE},A}(1^k, n(k)) < \varepsilon(k)$, where $\mathsf{Exp}^{\text{fs-msk}}_{fs-\mathsf{PRE},A}$ is defined in Experiment 6.

We now show that this notion in the sense of Definition 9 is trivially implied by fs-IND-CPA-1 security for fs-PRE in the sense of Definition 8.

Lemma 1. *If a fs-PRE scheme is fs-IND-CPA-1 secure in the sense of Definition 8, then the same fs-PRE scheme is fs-master-secret secure in the sense of Definition 9.*

[14] As we will discuss below, this notion seems to suggest false guarantees and leaves a critical gap in the security model open.

Proof sketch. It is trivial to see that any successful PPT adversary on the fs-master-secret security of a fs-PRE scheme can be transformed into a PPT adversary on the fs-IND-CPA-1 security of that fs-PRE scheme. (Essentially, any PPT adversary that is able to gain access to the secret key of the prior period can trivially distinguish ciphertexts for the same period.)

The problem with (fs-)PRE security. A problem with the notion of standard (i.e., IND-CPA and master secret) security for (plain) PRE and also our fs-PRE notions so far is that the secret keys used for level-1 (i.e., non re-encryptable) and level-2 (i.e., re-encryptable) ciphertexts can be independent. Consequently, although ciphertexts on both levels can be shown to be indistinguishable, this does not rule out the possibility that ciphertexts on level-2 reveal the respective level-2 secret key of the sender to an legitimate receiver. This is exactly the reason for the gap in the plain PRE model which allows to leak a "level-2 secret key" once a re-encryption has been performed while all security properties are still satisfied (we provide an example for such a scheme in Section 4.4). In particular, this allows the receiver to potentially decrypt *any* level-2 ciphertext. We provide a solution in form of a stronger security notion which we term fs-RIND-CPA security in the following.

fs-RIND-CPA security. We observe that existing PRE notions only consider that (1) as long as the users are honest, the proxy learns nothing about any plaintext, and (2) if proxies and users collude they do not learn anything about the ciphertexts which are not intended to be re-encrypted. We go a step further and consider malicious users in the face of an honest proxy in the forward-secret and, hence, also in the plain PRE sense. That is, we want to enforce that a malicious user can only read the ciphertexts which were actually re-encrypted by the proxy and can not tell anything about the ciphertexts which can potentially be re-encrypted. We capture this via the notion of fs-RIND-CPA security. In this scenario, an adversary receives re-encrypted ciphertexts generated by an honest proxy, that it is able to decrypt. Nevertheless, for all other level-2 ciphertexts, the adversary should still be unable to recover the plaintext. In Experiment 7, we model this notion where the adversary gets access to a ReEnc-oracle which is in possession of the re-encryption key from the target user to the adversary. We say A is valid if A only outputs $1 \leq j^* \leq n$ and equal length messages $|M_0| = |M_1|$.

```
Experiment $\mathsf{Exp}^{\mathsf{fs\text{-}rind\text{-}cpa}}_{fs-\mathsf{PRE},A}(1^k, n)$
    $\mathsf{pp} \leftarrow \mathsf{Setup}(1^k), (\mathsf{pk}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n), b \xleftarrow{R} \{0,1\}$
    $(j^*, \mathsf{pk}^*, \mathsf{st}) \leftarrow A(\mathsf{pp}, n, \mathsf{pk})$
    $(\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)}) \leftarrow \mathsf{Evo}(\mathsf{sk}^{(j-1)}, \mathsf{ek}^{(j-1)})$ for $j \in [j^*]$
    $\mathsf{rk} \leftarrow \mathsf{ReGen}(\mathsf{sk}^{(j^*)}, \perp, \mathsf{pk}^*)$
    $(M_0, M_1, \mathsf{st}) \leftarrow A^{\{\mathsf{ReEnc}(\mathsf{rk},\cdot)\}}(\mathsf{st})$
    $b^* \leftarrow A(\mathsf{st}, \mathsf{Enc}^{(2)}(\mathsf{pk}, M_b, j^*))$
    if $b = b^*$ return 1, else return 0
```

Experiment 7. The fs-RIND-CPA security experiment for fs-PRE schemes.

Definition 10 (fs-RIND-CPA). *For a polynomially bounded function $n(\cdot)$ and a PPT adversary A, we define the advantage function for A in the sense of fs-RIND-CPA as*

$$\mathsf{Adv}^{\text{fs-rind-cpa}}_{fs-\mathsf{PRE},A}(1^k, n(k)) := \left| \Pr\left[\mathsf{Exp}^{\text{fs-rind-cpa}}_{fs-\mathsf{PRE},A}(1^k, n(k)) = 1 \right] - \frac{1}{2} \right| .$$

*A fs-*PRE *scheme is fs-RIND-CPA if for all polynomially bounded $n(\cdot)$ and any valid PPT A there exists a negligible function ε such that $\mathsf{Adv}^{\text{fs-rind-cpa}}_{fs-\mathsf{PRE},A}($ $1^k, n(k)) < \varepsilon(k)$, where $\mathsf{Exp}^{\text{fs-rind-cpa}}_{fs-\mathsf{PRE},A}$ is defined in Experiment 7.*

We distinguish fs-PRE schemes based on this last notion:

Definition 11 (fs-PRE$^-$-security). *If a fs-*PRE *scheme is fs-IND-CPA-1 and fs-IND-CPA-2 secure, then we say this fs-*PRE *scheme is fs-*PRE$^-$*-secure.*

Definition 12 (fs-PRE$^+$-security). *If a fs-*PRE *scheme is fs-IND-CPA-1, fs-IND-CPA-2, and fs-RIND-CPA secure, then we say this fs-*PRE *scheme is fs-*PRE$^+$*-secure.*

3.3 Stronger Security for Proxy Re-Encryption

To conclude the discussion of the security model of fs-PRE schemes, we first observe that it is interesting to consider the notion of fs-RIND-CPA security in the classical setting for PRE, i.e., Experiment 7 with fixed $n = 1$ and no call to the Evo algorithm. The notion again ensures involvement of the proxy for the re-encryption of every ciphertext, and can, thus, enforce that malicious users cannot learn anything beyond the explicitly re-encrypted ciphertexts. This immediately leads to a stronger security model for classical PRE (given in the full version), which we denote as PRE$^+$. In particular, it extends the classical model [4], dubbed PRE$^-$, which covers standard (IND-CPA) and master-secret security definitions, by our fs-RIND-CPA security notion ported to the PRE setting. As our fs-IND-CPA-i notions for fs-PRE are generalizations of the established standard security notions of PRE as defined in [4], we consequently obtain a PRE$^+$-secure PRE scheme from any fs-PRE$^+$-secure fs-PRE scheme. We formalize this observation via Lemma 2.

Lemma 2. *Any fs-*PRE$^+$*-secure fs-*PRE *scheme yields a* PRE$^+$*-secure* PRE *scheme.*

In the full version, we formally prove this lemma.

4 Constructing fs-PRE from Binary Tree Encryption

In this section we present our construction of fs-PRE which is based on BTEs. Along the way, we introduce the notion of forward-secret delegatable PKE (fs-DPKE) as intermediate step. Such a fs-DPKE scheme then directly gives us a first fs-PRE satisfying fs-PRE$^-$ security. To extend our construction to satisfy the stronger fs-PRE$^+$ notion generically, we require a relatively mild homomorphic property of the fs-DPKE. This property is in particular satisfied by our fs-DPKE instantiation, which yields the first fs-PRE scheme with strong security.

4.1 Forward-Secret Delegatable Public-Key Encryption

We now formalize fs-DPKE. In such a scheme decryption rights within a public-key encryption scheme can be delegated from a delegator to a delegatee and secret keys of delegators can be evolved so that a secret key for some period e_i is no longer useful to decrypt ciphertexts of prior periods e_j with $j < i$.

Definition 13 (fs-DPKE). *A forward-secret delegatable* PKE *(fs-*DPKE*) scheme with message space $\mathcal{M}$ consists of the PPT algorithms* (Setup, Gen, Evo, Del, Enc, Dec, DelEvo, DelDec) *as follows:*

$\mathsf{Setup}(1^k)$: *On input security parameter k, outputs public parameters* pp*.*

$\mathsf{Gen}(\mathsf{pp}, n)$: *On input public parameters* pp*, and maximum number of periods n, outputs public, secret and evolution keys* $(\mathsf{pk}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)})$*.*

$\mathsf{Evo}(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})$: *On input secret key $\mathsf{sk}^{(i)}$, and evolution key $\mathsf{ek}^{(i)}$ for period $i \in \{0, \ldots, n-2\}$, outputs secret key $\mathsf{sk}^{(i+1)}$ and evolution key $\mathsf{ek}^{(i+1)}$ for period $i+1$.*

$\mathsf{Del}(\mathsf{sk}_A^{(i)}, \mathsf{ek}_A^{(i)}, \mathsf{pk}_B)$: *On input secret key $\mathsf{sk}_A^{(i)}$ and evolution key $\mathsf{ek}_A^{(i)}$ (or $\perp$) for A and period $i \in \{0, \ldots, n-1\}$, and public key pk_B for B, outputs delegated key $\mathsf{dk}^{(i)}$ and delegated evolution key $\mathsf{dek}^{(i)}$ (or $\perp$) for period i.*

$\mathsf{Enc}(\mathsf{pk}, M, i)$: *On input a public key* pk*, a message $M \in \mathcal{M}$, and period $i \in \{0, \ldots, n-1\}$, outputs a ciphertext C.*

$\mathsf{Dec}(\mathsf{sk}^{(i)}, C)$: *On input a secret key $\mathsf{sk}^{(i)}$, for period $i \in \{0, \ldots, n-1\}$, and ciphertext C, outputs $M \in \mathcal{M} \cup \{\perp\}$.*

$\mathsf{DelEvo}(\mathsf{dk}^{(i)}, \mathsf{dek}^{(i)})$: *On input a delegation key $\mathsf{dk}^{(i)}$ and delegated evolution key $\mathsf{dek}^{(i)}$ for period $i \in \{0, \ldots, n-2\}$, output delegation key $\mathsf{dk}^{(i+1)}$ and delegated evolution key $\mathsf{dek}^{(i+1)}$ for period $i+1$.*

$\mathsf{DelDec}(\mathsf{sk}_B^{(i)}, \mathsf{dk}_{A\to B}^{(i)}, C_A)$: *On input secret key $\mathsf{sk}_B^{(i)}$ for B and period $i \in \{0, \ldots, n-1\}$, delegation key $\mathsf{dk}_{A\to B}^{(i)}$ from A for B and period i, and ciphertext C_A for A, outputs $M \in \mathcal{M} \cup \{\perp\}$.*

We note that the existence of the DelEvo algorithm is entirely optional. If provided, it allows the user in possession of a delegation key to evolve it for later periods without additional interaction with the delegator.

Correctness. For correctness we require that period i ciphertexts encrypted for user u can be decrypted if one is in possession of the secret key of u evolved to that period or one possess a delegation key of u to another user u' and the secret key for u' for that period. More formally, we require that for all security parameters $k \in \mathbb{N}$, all public parameters pp generated by $\mathsf{Setup}(1^k)$, all number of periods $n \in \mathbb{N}$, all users $U \in \mathbb{N}$, all key tuples $(\mathsf{pk}_u, \mathsf{sk}_u^{(0)}, \mathsf{ek}_u^{(0)})_{u\in[U]}$ generated by $\mathsf{Gen}(\mathsf{pp}, n)$, any period $i \in \{0, \ldots, n-1\}$, for any $u \in [U]$, any evolved keys $(\mathsf{sk}_u^{(i)}, \mathsf{ek}_u^{(i)})$ generated by Evo from $(\mathsf{sk}_u^{(0)}, \mathsf{ek}_u^{(0)})$, for all $u' \in [U], u \neq u'$, any (potentially evolved) delegation key $\mathsf{dk}_{u \to u'}^{(i)}$ for period i generated using Del from a (potentially evolved) secret key and the target public key, and all messages $M \in \mathcal{M}$ it holds that

$$\mathsf{Dec}(\mathsf{sk}_u^{(i)}, \mathsf{Enc}(\mathsf{pk}_u, M, i)) = \mathsf{DelDec}(\mathsf{sk}_{u'}^{(i)}, \mathsf{dk}_{u \to u'}^{(i)}, \mathsf{Enc}(\mathsf{pk}_u, M, i)) = M.$$

Security notions. The forward-secret IND-CPA notion is a straight-forward extension of the typical IND-CPA notion: the adversary selects a target period and gets access to secret and evolution keys of the targeted user for the selected period and is able to request delegation keys with honest and dishonest users for that period. The adversary then engages with an IND-CPA style challenge for the previous period. For the experiment, which is depicted in Experiment 8, the environment keeps a list of an initial empty list of honest users $\mathtt{HU}$.

$\mathsf{Gen}^{(h)}(\mathsf{pp}, n)$: Run $(\mathsf{pk}, \mathsf{sk}, \mathsf{ek}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n)$, set $\mathtt{HU} \leftarrow \mathtt{HU} \cup \{(\mathsf{pk}, \mathsf{sk}, \mathsf{ek})\}$, and return pk.

$\mathsf{Del}^{(h)}(j, \mathsf{pk}_u, \mathsf{pk})$: On input a period j, a public key pk_u and a public key pk, abort if $(\mathsf{pk}_u, \cdot) \notin \mathtt{HU}$. Otherwise, look up $\mathsf{sk}_u^{(0)}, \mathsf{ek}_u^{(0)}$ corresponding to pk_u from $\mathtt{HU}$, set $(\mathsf{sk}_u^{(i)}, \mathsf{ek}_u^{(i)}) \leftarrow \mathsf{Evo}(\mathsf{sk}_u^{(i-1)}, \mathsf{ek}_u^{(i-1)})$ for $i \in [j]$ if $j > 0$, and return $\mathsf{Del}(\mathsf{sk}_u^{(j)}, \mathsf{ek}_u^{(j)}, \mathsf{pk})$.

$\mathsf{Del}^{(h')}(j, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \mathsf{pk}_u)$: On input a period j, a secret key $\mathsf{sk}^{(0)}$, a evolution key $\mathsf{ek}^{(0)}$, and a public key pk_u, abort if $(\mathsf{pk}_u, \cdot) \notin \mathtt{HU}$. Otherwise, set $(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)}) \leftarrow \mathsf{Evo}(\mathsf{sk}^{(i-1)}, \mathsf{ek}^{(i-1)})$ for $i \in [j]$ if $j > 0$, and return $\mathsf{Del}(\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)}, \mathsf{pk}_u)$.

```
Experiment $\mathsf{Exp}^{\mathsf{fs\text{-}ind\text{-}cpa}}_{fs-\mathsf{DPKE},A}(1^k, n)$
  $\mathsf{pp} \leftarrow \mathsf{Setup}(1^k), (\mathsf{pk}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n), b \xleftarrow{R} \{0,1\}$
  $(j^*, \mathsf{st}) \leftarrow A(\mathsf{pp}, n, \mathsf{pk})$
  $\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)} \leftarrow \mathsf{Evo}(\mathsf{sk}^{(j-1)}, \mathsf{ek}^{(j-1)})$ for $j \in [j^*]$.
  $\mathcal{O} \leftarrow \{\mathsf{Gen}^{(h)}, \mathsf{Del}^{(h)}(j^*-1, \cdot, \mathsf{pk}), \mathsf{Del}^{(h')}(j^*-1, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \cdot)\}$
  $(M_0, M_1, \mathsf{st}) \leftarrow A^{\mathcal{O}}(\mathsf{st}, \mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)}))$
  $b^* \leftarrow A(\mathsf{st}, \mathsf{Enc}(\mathsf{pk}, M_b, j^*-1))$
  if $b = b^*$ return 1, else return 0
```

Experiment 8. The fs-IND-CPA security experiment for a fs-DPKE scheme.

Definition 14 (fs-IND-CPA). *For a polynomially bounded function $n(\cdot) > 1$, a PPT adversary A, we define the advantage function in the sense of fs-IND-CPA as*

$$\mathsf{Adv}^{\mathsf{fs\text{-}ind\text{-}cpa}}_{fs-\mathsf{DPKE},A}(1^k, n(k)) := \left| \Pr\left[\mathsf{Exp}^{\mathsf{fs\text{-}ind\text{-}cpa}}_{fs-\mathsf{DPKE},A}(1^k, n(k)) = 1 \right] - \frac{1}{2} \right|.$$

If for all $n(\cdot) > 1$, and any A there exists a negligible function ε such that $\mathsf{Adv}^{\mathsf{fs\text{-}ind\text{-}cpa}}_{fs-\mathsf{DPKE},A}(1^k, n(k)) < \varepsilon(k)$, then a fs-DPKE scheme is fs-IND-CPA secure.

4.2 Constructing fs-DPKE from BTE

Now we construct a fs-DPKE scheme from a BTE scheme by applying the CHK compiler to a BTE and combining it with an $\mathcal{F}$-HPKE scheme for handling the delegation keys, i.e., the fs-DPKE key contains a BTE and an $\mathcal{F}$-HPKE key. The evolution key contains the secret and derivation keys for all right siblings on the path from the root node to w^i as well as the evolution key for w^i. The evolution algorithms traverse the tree in a depth-first manner, hence the evolution keys are viewed as stack and when visiting a node, the derived secret and derivation keys are pushed onto the stack. To simplify the presentation of the scheme, we define an algorithm DFEval that performs the stack manipulation on a stack of pairs:

$\mathsf{DFEval}(s_1^{(w^i)}, s, \mathsf{Eval})$: On input the stack s and first element $s_1^{(w^i)}$ of the pair for node w^i, an algorithm Eval, perform the following steps:

- Pop the topmost element, $(\perp, s_2^{(w^i)})$, from the stack s.
- If w^i is an internal node, set $s^{(w^i0)}, s^{(w^i1)} \leftarrow \mathsf{Eval}(s_1^{(w^i)}, s_2^{(w^i)})$ and push $s^{(w^i1)}, s^{(w^i0)}$ onto s.
- Replace the topmost element, $(s_1^{(w^{i+1})}, s_2^{(w^{i+1})})$, with $(\perp, s_2^{(w^{i+1})})$.
- Return $s_1^{(w^{i+1})}$ and the new stack s.

The overall idea is now to encrypt the BTE secret key of the current period using the $\mathcal{F}$-HPKE scheme's public key of the target user. Using the homomorphic property of the encryption scheme, we are able to evolve the delegation keys in the same way as the secret keys of the nodes. In particular, we will require that the key derivation algorithm of the BTE can be represented by functions in $\mathcal{F}$, i.e., $\mathsf{Der}_{\mathsf{BTE}} = (f_i)_{i\in[m]}$. For notional simplicity, we will write $\mathsf{Eval}_{\mathsf{HPKE}}(\mathsf{Der}_{\mathsf{BTE}}, \cdot)$ instead of repeating it for each f_i that represents $\mathsf{Der}_{\mathsf{BTE}}$.

For our fs-DPKE scheme we need keys of different users to live in compatible key spaces. To that end, we introduce Setup algorithms for both schemes that fix the key spaces and we change the key generation algorithms to take the public parameters instead of the security parameter as argument. Note that when using the BTE from [15], linear ElGamal [11] as $\mathcal{F}$-HPKE to encrypt the BTE keys suffices for our needs.

Our construction. The fs-DPKE scheme is detailed in Scheme 1. We note that only the definition of DelEvo relies on the homomorphic properties of the HPKE scheme. So to obtain a fs-DPKE scheme without DelEvo algorithm, a compatible PKE scheme is sufficient. Yet, we will require the homomorphic properties later to achieve a suitable notion of adaptability regardless of the availability of DelEvo.

Similar to Canetti et al.'s construction, our fs-DPKE scheme inherits the fs-IND-CPA security from the BTE's IND-SN-CPA security.

Theorem 1. *If instantiated with an IND-SN-CPA secure* BTE *scheme and a IND-CPA secure* HPKE *scheme, then Scheme 1 is a fs-IND-CPA secure fs-*DPKE*.*

Let $(\mathsf{Setup}_{\mathsf{BTE}}, \mathsf{Gen}_{\mathsf{BTE}}, \mathsf{Der}_{\mathsf{BTE}}, \mathsf{Enc}_{\mathsf{BTE}}, \mathsf{Dec}_{\mathsf{BTE}})$ be a BTE scheme and $(\mathsf{Setup}_{\mathsf{HPKE}}, \mathsf{Gen}_{\mathsf{HPKE}}, \mathsf{Enc}_{\mathsf{HPKE}}, \mathsf{Dec}_{\mathsf{HPKE}}, \mathsf{Eval}_{\mathsf{HPKE}})$ a compatible $\mathcal{F}$-HPKE scheme with $\mathsf{Der}_{\mathsf{BTE}} \in \mathcal{F}$.

$\underline{\mathsf{Setup}(1^k)}$: Set $\mathsf{pp}_{\mathsf{BTE}} \leftarrow \mathsf{Setup}_{\mathsf{BTE}}(1^k)$, $\mathsf{pp}_{\mathsf{HPKE}} \leftarrow \mathsf{Setup}_{\mathsf{HPKE}}(1^k)$, and return $(\mathsf{pp}_{\mathsf{BTE}}, \mathsf{pp}_{\mathsf{HPKE}})$.

$\underline{\mathsf{Gen}(\mathsf{pp}, n)}$: Parse pp as $(\mathsf{pp}_{\mathsf{BTE}}, \mathsf{pp}_{\mathsf{HPKE}})$. Choose ℓ such that $n < 2^{\ell+1}$, set $(\mathsf{pk}_{\mathsf{BTE}}, \mathsf{sk}_{\mathsf{BTE}}^{(\varepsilon)}, \mathsf{dk}_{\mathsf{BTE}}^{(\varepsilon)}) \leftarrow \mathsf{Gen}_{\mathsf{BTE}}(\mathsf{pp}_{\mathsf{BTE}}, \ell)$ and $(\mathsf{pk}_{\mathsf{HPKE}}, \mathsf{sk}_{\mathsf{HPKE}}) \leftarrow \mathsf{Gen}_{\mathsf{HPKE}}(\mathsf{pp}_{\mathsf{HPKE}})$, and return $((\mathsf{pk}_{\mathsf{BTE}}, \mathsf{pk}_{\mathsf{HPKE}}), (\mathsf{sk}_{\mathsf{BTE}}^{(\varepsilon)}, \mathsf{sk}_{\mathsf{HPKE}}), (\perp, \mathsf{dk}_{\mathsf{BTE}}^{(\varepsilon)}))$.

$\underline{\mathsf{Evo}(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})}$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}_{\mathsf{BTE}}^{(w^i)}, \mathsf{sk}_{\mathsf{HPKE}})$ and view $\mathsf{ek}^{(i)}$ organized as a stack of secret key and evolution keys pairs. Set $\mathsf{sk}_{\mathsf{BTE}}^{(w^{i+1})}, \mathsf{ek}^{(i+1)} \leftarrow \mathsf{DFEval}(\mathsf{sk}_{\mathsf{BTE}}^{(w^i)}, \mathsf{ek}^{(i)}, \mathsf{Der}_{\mathsf{BTE}})$, and $\mathsf{sk}^{(i+1)} \leftarrow (\mathsf{sk}_{\mathsf{BTE}}^{(w^{i+1})}, \mathsf{sk}_{\mathsf{HPKE}})$. Return $\mathsf{sk}^{(i+1)}, \mathsf{ek}^{(i+1)}$.

$\underline{\mathsf{Enc}(\mathsf{pk}, M, i)}$: Parse pk as $(\mathsf{pk}_{\mathsf{BTE}}, \cdot)$, and return $\mathsf{Enc}_{\mathsf{BTE}}(\mathsf{pk}_{\mathsf{BTE}}, M, w^i)$.

$\underline{\mathsf{Dec}(\mathsf{sk}^{(i)}, C)}$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}_{\mathsf{BTE}}^{(w^i)}, \cdot)$, and return $\mathsf{Dec}_{\mathsf{BTE}}(\mathsf{sk}_{\mathsf{BTE}}^{(w^i)}, C)$.

$\underline{\mathsf{Del}(\mathsf{sk}_A^{(i)}, \mathsf{ek}_A^{(i)}, \mathsf{pk}_B)}$: Parse $\mathsf{sk}_A^{(i)}$ as $(\mathsf{sk}_{\mathsf{BTE}}^{(w^i)}, \cdot)$ and pk_B as $(\cdot, \mathsf{pk}_{\mathsf{HPKE}})$. If $\mathsf{ek}_A^{(i)} = \perp$, return $\mathsf{Enc}_{\mathsf{HPKE}}(\mathsf{pk}_{\mathsf{HPKE}}, \mathsf{sk}_{\mathsf{BTE}}^{(w^i)})$. Otherwise parse $\mathsf{ek}_A^{(i)}$ as $(\mathsf{sk}_{\mathsf{BTE}}^{(w)}, \mathsf{dk}_{\mathsf{BTE}}^{(w)})_{w \in W}, (\cdot, \mathsf{dk}_{\mathsf{BTE}}^{(w^i)})$, and set $\mathsf{dk}^{(w)} \leftarrow \mathsf{Enc}_{\mathsf{HPKE}}(\mathsf{pk}_{\mathsf{HPKE}}, \mathsf{sk}_{\mathsf{BTE}}^{(w)})$ and $\mathsf{dek}^{(w)} \leftarrow \mathsf{Enc}_{\mathsf{HPKE}}(\mathsf{pk}_{\mathsf{HPKE}}, \mathsf{dk}_{\mathsf{BTE}}^{(w)})$ for $w \in W \cup \{w^i\}$. Set $\mathsf{dk}^{(i)} \leftarrow \mathsf{dk}^{(w^i)}$ and $\mathsf{dek}^{(i)} \leftarrow (\mathsf{dk}^{(w)}, \mathsf{dek}^{(w)})_{w \in W}, (\perp, (\mathsf{dek}^{(w^i)}))$ and return $\mathsf{dk}^{(i)}, \mathsf{dek}^{(i)}$.

$\underline{\mathsf{DelEvo}(\mathsf{dk}_{A \to B}^{(i)}, \mathsf{dek}_{A \to B}^{(i)})}$: Parse $\mathsf{dk}_{A \to B}^{(i)}$ as $\mathsf{dk}_{A \to B}^{(w^i)}$ and view $\mathsf{dek}_{A \to B}^{(i)}$ organized as a stack of encrypted evolution keys. Set $\mathsf{dk}_{A \to B}^{(w^{i+1})}, \mathsf{dek}_{A \to B}^{(i+1)} \leftarrow \mathsf{DFEval}(\mathsf{dk}_{A \to B}^{(w^i)}, \mathsf{dek}_{A \to B}^{(i)}, \mathsf{Eval}_{\mathsf{HPKE}}(\mathsf{Der}_{\mathsf{BTE}}, \cdot))$, and $\mathsf{dk}^{(i+1)} \leftarrow \mathsf{dk}_{\mathsf{BTE}}^{(w^{i+1})}$. Return $\mathsf{dk}^{(i+1)}, \mathsf{dek}^{(i+1)}$.

$\underline{\mathsf{DelDec}(\mathsf{sk}_B^{(i)}, \mathsf{dk}_{A \to B}^{(i)}, C_A)}$: Parse $\mathsf{sk}_B^{(i)}$ as $(\cdot, \mathsf{sk}_{\mathsf{HPKE}})$, set $\mathsf{sk}_{\mathsf{BTE}}^{(w^i)} \leftarrow \mathsf{Dec}_{\mathsf{HPKE}}(\mathsf{sk}_{\mathsf{HPKE}}, \mathsf{dk}_{A \to B}^{(i)})$, and return $\mathsf{Dec}_{\mathsf{BTE}}(\mathsf{sk}_{\mathsf{BTE}}^{(w^i)}, C_A)$.

Scheme 1. fs-DPKE scheme from BTE scheme and a compatible HPKE scheme.

Proof. We prove the theorem using a sequence of games. We denote by W all the relevant nodes in the binary tree for period j. We note that the size of W is bounded by $\log_2(n)$. We index W as w_i for $i \in [|W|]$.

Game 0: The original game.

Game $1_{i,j}$ $(1 \leq i \leq q_{\mathsf{Del}^h}, 1 \leq j \leq 2|W|)$: As the previous game, but we replace all HPKE ciphertexts up to the j-th one in the i-th query with ciphertexts encrypting random plaintexts. That is, we modify the $\mathsf{Del}^{h'}$ in the i-th query as follows:

$\mathsf{Del}^{h'}(j, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \mathsf{pk}_i)$: Up to the j-th call to $\mathsf{Enc}_{\mathsf{HPKE}}$, encrypt a uniformly random value.

Transition$^{0 \to 1_{1,1}}$, Transition$^{1_{i,j} \to 1_{i,j+1}}$, Transition$^{1_{i,2|W|} \to 1_{i+1,1}}$: A distinguisher $\mathcal{D}^{0 \to 1_{1,1}}$ (respectively $\mathcal{D}^{1_{i,j} \to 1_{i,j+1}}$ or $\mathcal{D}^{1_{i,2|W|} \to 1_{i+1,1}}$) is an IND-CPA adversary against the HPKE scheme. We construct a reduction where we let $\mathcal{C}$ be a IND-CPA challenger. We modify $\mathsf{Del}^{h'}$ in the i-th query in the following way:

$\mathsf{Del}^{h'}(j, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \mathsf{pk}_{i'})$: Simulate everything honestly, but on the j-th query choose $\boxed{r}$ uniformly at random and run

$$\boxed{c \leftarrow \mathcal{C}(\mathsf{sk}_{BTE}^{(w_{(j/2)-1})}, r)} \text{ if } j \text{ is odd and } \boxed{c \leftarrow \mathcal{C}(\mathsf{ek}_{BTE}^{(w_{j/2})}, r)} \text{ if } j \text{ is even,}$$

where $c \leftarrow \mathcal{C}(m_0, m_b)$ denotes a challenge ciphertext with respect to m_0 and m_1.

Now, the bit b chosen by $\mathcal{C}$ switches between the distributions of the Games.

In Game $1_{q_{\mathsf{Del}^h}, 2|W|}$ all ciphertexts obtainable from $\mathsf{Del}^{h'}$ are with respect to random values. Now, an adversary B winning Game $1_{q_{\mathsf{Del}^h}, 2|W|}$ can be transformed into a IND-SN-CPA adversary A against the underlying BTE scheme:

1. When A is first started on $1^k, \ell$, choose $i^* \xleftarrow{R} [n]$ and output $w^{(i^*-1)}$.
2. When A is started on $\mathsf{pk}_{\mathsf{BTE}}, (\mathsf{sk}^{(w)}, \mathsf{dk}^{(w)})_{w \in W}$, compute $(\mathsf{pk}_{\mathsf{HPKE}}, \mathsf{sk}_{\mathsf{HPKE}}) \leftarrow \mathsf{Gen}_{\mathsf{HPKE}}(1^k)$. The secret key $\mathsf{sk}_{\mathsf{HPKE}}$ is stored in the state st and we extend the public key to $\mathsf{pk} \leftarrow (\mathsf{pk}_{\mathsf{BTE}}, \mathsf{pk}_{\mathsf{HPKE}})$. Now start B on the extended public key, i.e. $(j^*, \mathsf{st}) \leftarrow B(1^k, n, \mathsf{pk})$. If $i^* \neq j^*$, output a random bit and halt. Otherwise we have the secret-derivation key pairs of all nodes that are right siblings on the path from the root node to $w^{(j^*-1)}$ and (if they exist) all child nodes of $w^{(j^*-1)}$, hence we are able to simulate all oracle queries from B honestly. Similarly, we can compute $(\mathsf{sk}^{(j^*)}, \mathsf{dk}^{(j^*)})$ from the given keys. Thus we run $B^{\mathcal{O}}(\mathsf{st}, \mathsf{sk}^{(j^*)}, \mathsf{dk}^{(j^*)})$ and forward its result.
3. When A is finally started on the challenge ciphertext, the ciphertext is simply forwarded to B and when B outputs the bit b, A returns b and halts.

When B is running within A and $j^* = i^*$, B has exactly the same view as in Game $1_{q_{\mathsf{Gen}^h}, 2|W|}$. In this case the probability of A to win is exactly the same as the winning probability of B, and Game $1_{q_{\mathsf{Gen}^h}, 2|W|}$ is computationally indistinguishable from the initial game. The random guess of i^* so that $i^* = j^*$ induces a loss of $\frac{1}{n}$, which is however bounded by a polynomial in the security parameter. □

4.3 Constructing fs-PRE from fs-DPKE

Now we present a construction of a fs-PRE$^+$-secure fs-PRE scheme from a fs-DPKE scheme. Therefore, we define additional properties of fs-DPKE and show that a fs-PRE can be directly obtained from a fs-DPKE. For our transformation to work, we need to define an additional algorithm that allows us to homomorphically shift ciphertexts and delegation keys. That is, ciphertexts and delegation keys are modified in such a way that the delegation keys look like randomly distributed fresh keys, which are only useful to decrypt ciphertexts adapted to this key. Formally, we introduce an algorithm Adapt that enables this adaption:

$\mathsf{Adapt}(\mathsf{dk}, C)$: On input a delegation key dk, a ciphertext C, outputs an adapted delegation key dk' and ciphertext C'.

Since the delegation keys in our construction are encrypted BTE secret keys, we essentially adapt secret keys and ciphertexts from a BTE. We will see that this adaption is possible as long as the HPKE scheme used to encrypt the BTE keys provides a suitable homomorphism on the message space.

To adapt ciphertexts and delegation keys we extend correctness to additionally require that for any message M encrypted under the public key of A, any delegation key $\mathsf{dk}_{A \to B}^{(i)}$, and any adapted delegation key-ciphertext pairs $(\mathsf{dk}', C') \leftarrow \mathsf{Adapt}(\mathsf{dk}_{A \to B}^{(i)}, C_A)$, it holds that $M = \mathsf{DelDec}_{\mathsf{DPKE}}(sk_B^{(i)}, \mathsf{dk}', C')$.

As security notion we introduce the fs-ADAP-IND-CPA notion, where the adversary may see multiple adapted delegation keys and ciphertexts, but the adversary should be unable to win an IND-CPA game for non-adapted ciphertexts. We give the formal definition of the security experiment in Experiment 9. This notion gives the delegator more control over the ciphertexts that should be readable for the delegatee. If given the delegation key, the delegatee can always decrypt all ciphertexts, but if just given an adapted delegation key, only a selected subset of ciphertexts is decryptable.

```
Experiment $\mathsf{Exp}^{\text{fs-adap-ind-cpa}}_{fs-\mathsf{DPKE},A}(1^k, n)$
  $\mathsf{pp} \leftarrow \mathsf{Setup}(1^k), (\mathsf{pk}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n), b \xleftarrow{R} \{0, 1\}$
  $(j^*, \mathsf{pk}^*, \mathsf{st}) \leftarrow A(\mathsf{pp}, n, \mathsf{pk})$
  $\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)} \leftarrow \mathsf{Evo}(\mathsf{sk}^{(j-1)}, \mathsf{ek}^{(j-1)})$ for $j \in [j^*]$, $\mathsf{dk} \leftarrow \mathsf{Del}(\mathsf{sk}^{(j^*)}, \perp, \mathsf{pk}^*)$
  $(M_0, M_1, \mathsf{st}) \leftarrow A^{\{\mathsf{Adapt}(\mathsf{dk}, \cdot)\}}(\mathsf{st})$
  $b^* \leftarrow A(\mathsf{st}, \mathsf{Enc}(\mathsf{pk}, M_b, j^*))$
  if $b = b^*$ return 1, else return 0
```

Experiment 9. The fs-ADAP-IND-CPA security experiment for a fs-DPKE scheme.

Definition 15 (fs-ADAP-IND-CPA). *For a polynomially bounded function $n(\cdot) > 1$, a PPT adversary A, we define the advantage function in the sense of fs-IND-CPA as*

$$\mathsf{Adv}^{\text{fs-adap-ind-cpa}}_{fs-\mathsf{DPKE},A}(1^k, n(k)) := \left| \Pr\left[\mathsf{Exp}^{\text{fs-adap-ind-cpa}}_{\mathsf{DPKE},A}(1^k, n(k)) = 1 \right] - \frac{1}{2} \right|.$$

If for all $n(\cdot) > 1$, and any A there exists a negligible function ε such that $\mathsf{Adv}^{\text{fs-adap-ind-cpa}}_{fs-\mathsf{DPKE},A}(1^k, n(k)) < \varepsilon(k)$, then a fs-$\mathsf{DPKE}$ scheme is fs-ADAP-IND-CPA secure.

For Scheme 1, this adaption can be achieved solely from key-homomorphic properties of the BTE and homomorphic properties of the HPKE, respectively. Subsequently, we define the required homomorphisms. Our definitions are inspired by [2, 40]. We focus on schemes where the secret/derived key pairs, and public keys live in groups $(\mathbb{G}, +)$, and $(\mathbb{H}, \cdot)$, respectively. We will require two different properties: first, the public key is the image of the secret key under a group homomorphism, and second, given two secret keys with a known difference, we can map the binary tree of derived keys from one key to the other key. In other words, the difference in the keys propagates to the derived keys.

Definition 16. *Let Ω be a* BTE *scheme with secret/derived key space* $(\mathbb{G}, +)$ *and public key space* $(\mathbb{H}, \cdot)$*.*

1. *The scheme Ω provides a secret-key-to-public-key homomorphism, if there exists an efficiently computable group homomorphism $\mu : \mathbb{G} \to \mathbb{H}$ such that for all* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$*, it holds that* $\mathsf{pk} = \mu(\mathsf{sk})$*.*
2. *The scheme Ω provides a derived-key homomorphism, if there exists a family of efficiently computable group homomorphisms $\nu^{(w)} : \mathbb{G} \to \mathbb{G}^2$ such that for all* $(\mathsf{pk}, \mathsf{sk}^{(\varepsilon)}) \leftarrow \mathsf{Gen}$*, all nodes w it holds that* $(\mathsf{sk}^{(w0)}, \mathsf{sk}^{(w1)}) = \nu^{(w)}(\mathsf{sk}^{(w)})$ *and for all messages M it holds that* $\mathsf{Dec}(\mathsf{sk}^{(w)}, \mathsf{Enc}(\mathsf{pk}, M, w)) = M$*.*

We denote by Φ^+ the set of all possible secret key differences in $\mathbb{G}$. Alternatively, it is possible to view Φ^+ as set of functions representing all linear shifts in $\mathbb{G}$ and we simply identify each shift by an element $\Delta \in \mathbb{G}$.

Definition 17. *A* BTE *scheme Ω is called Φ^+-key-homomorphic, if it provides both a secret-key-to-public-key homomorphism and a derived key homomorphism and an additional PPT algorithm* Adapt*, defined as:*

$\mathsf{Adapt}(\mathsf{pk}, C, \Delta)$: *On input a delegation key* dk*, a ciphertext C and a secret key difference Δ, outputs a public key* pk' *and a ciphertext C'.*

such that for all $\Delta \in \Phi^+$, and all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\ldots)$*, all message M, and all* $C \leftarrow \mathsf{Enc}(\mathsf{pk}, M)$*, and* $(\mathsf{pk}', C') \leftarrow \mathsf{Adapt}(\mathsf{pk}, C, \Delta)$ *it holds that* $\mathsf{pk}' = \mathsf{pk} \cdot \mu(\Delta)$ *and* $\mathsf{Dec}(\mathsf{sk}^{(w)} + \nu^{(w)}(\Delta), C') = M$*.*

Definition 18 (Adaptability of ciphertexts). *A Φ^+-key-homomorphic* BTE *scheme provides adaptability of ciphertexts, if for every security parameter $k \in \mathbb{N}$, any public parameters* $\mathsf{pp} \leftarrow \mathsf{Setup}(1^k)$*, every message M and every period j, it holds that* $\mathsf{Adapt}(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, M, j), \Delta)$ *and* $(\mathsf{pk} \cdot \mu(\Delta), \mathsf{Enc}(\mathsf{pk} \cdot \mu(\Delta), M, j))$ *as well as* $(\mathsf{sk}, \mathsf{pk})$ *and* $(\mathsf{sk}', \mu(\mathsf{sk}'))$ *are identically distributed, where* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n)$*,* $\mathsf{sk}' \overset{R}{\leftarrow} \mathbb{G}$ *and* $\Delta \leftarrow \Phi^+$*.*

Next, we discuss the BTE from [15] with respect to our notion of ciphertext adaptability. We first recall the BTE scheme in Scheme 2 where BGGen is a bilinear group generator. By [15, Proposition 1] this scheme is IND-SN-CPA secure if the decisional BDH assumption holds relative to BGGen.

Now we show that Scheme 2 also provides adaptability of ciphertexts:

Lemma 3. *Scheme 2 provides adaptability of ciphertexts under shared H.*

Proof. We show the existence of the homomorphisms and give the Adapt algorithm. Note that the master secret key can easily be viewed as containing α, hence, the secret-to-public-key homomorphism is simply $\mu : \alpha \mapsto \alpha P$. As the Der algorithm simply computes sums, the existence of the homomorphism is clear.

We now show the existence of Adapt:

$\mathsf{Adapt}(\mathsf{pk}, C, \Delta)$: Parse pk as (Q, ℓ, H) and C as $(U_0, \ldots, U_t, V)$. Let $Q' \leftarrow Q + \Delta \cdot P$ and set $\mathsf{pk}' \leftarrow (Q', \ell, H)$. Let $V' \leftarrow Ve(U_0, \Delta \cdot H(\varepsilon))$ and set $C' \leftarrow (U_0, \ldots, U_t, V')$ and return (pk', C').

$\mathsf{Setup}(1^k)$: Run to $\mathsf{BGGen}_p(1^k)$ to generate groups $\mathbb{G}_1, \mathbb{G}_2$ of prime order q and a bilinear map e and select a random generator $P \in \mathbb{G}_1$. Set $\mathsf{pp} \leftarrow (\mathbb{G}_1, \mathbb{G}_2, e, q, P)$ and return pp.

$\mathsf{Gen}(\mathsf{pp}, \ell)$: Choose $\alpha \leftarrow \mathbb{Z}_q$ and set $Q \leftarrow \alpha \cdot P$. Set $\mathsf{sk}^{(\varepsilon)} \leftarrow \alpha H(\varepsilon)$ and $\mathsf{pk} \leftarrow (Q, H)$. Return $(\mathsf{pk}, \mathsf{sk}^{(\varepsilon)})$.

$\mathsf{Der}(\mathsf{sk}^{(i)})$: Parse $\mathsf{sk}^{(w)}$ as $(R_{w|1}, \ldots, R_w, S_w)$. Choose $r_0, r_1 \xleftarrow{R} \mathbb{Z}_q$ and set $R_{wi} \leftarrow r_i P$ and $S_{wi} \leftarrow S_w + r_i \cdot H(wi)$ for $i \in [2]$ and return $((R_{w|1}, \ldots, R_w, R_{w0}, S_{w0}), (R_{w|1}, \ldots, R_w, R_{w1}, S_{w1})))$.

$\mathsf{Enc}(\mathsf{pk}, M, i)$: Choose $\gamma \leftarrow \mathbb{Z}_q$ and set $C \leftarrow (\gamma \cdot P, \gamma \cdot H(w|1), \ldots, \gamma \cdot H(w), M \cdot e(Q, \gamma \cdot H(\varepsilon)))$. Return C.

$\mathsf{Dec}(\mathsf{sk}^{(w)}, C)$: Parse $\mathsf{sk}^{(w)}$ as $(R_{w|1}, \ldots, R_w, S_w)$ and $\mathcal{C}$ as $(U_0, \ldots, U_t, V)$. Return $M = V/d$ where

$$d = \frac{e(U_0, S_w)}{\prod_{i=1}^{t} e(R_{w|i}, U_i)}.$$

Scheme 2. BTE scheme from [15]

The adapted C' ciphertext is an encryption of the original message under the public key $Q' = Q + \varDelta \cdot P$. □

Now, given any $\varPhi^+$-key-homomorphic BTE scheme, it can be turned into an adaptable fs-DPKE by defining Adapt in a publicly computable way as follows:

$\mathsf{Adapt}(\mathsf{dk}^{(i)}_{A\to B}, C)$: Sample $\varDelta \xleftarrow{R} \varPhi^+$ and compute $\mathsf{dk}_\varDelta \leftarrow \mathsf{Enc}_{\mathsf{HPKE}}(\mathsf{pk}_B, \nu^{(w^i)}($ $\varDelta))$, and then $\mathsf{dk}' \leftarrow \mathsf{Eval}_{\mathsf{HPKE}}(+, \mathsf{dk}^{(i)}_{A\to B}, \mathsf{dk}_\varDelta)$. Set $(\cdot, C') \leftarrow \mathsf{Adapt}_{\mathsf{BTE}}(\mathsf{pk}_A,$ $C, \varDelta)$. Return (dk', C').

Theorem 2. *If in addition to the premise in Theorem 1 the* BTE *scheme also provides adaptability of ciphertexts, then Scheme 1 is a fs-ADAP-IND-CPA secure fs-*DPKE *scheme.*

Proof. We prove this theorem with a sequence of games.

Game 0: The original game.

Game 1: We modify the simulation of the Adapt oracle as follows, where we denote the modified oracle by Adapt':

$\mathsf{Adapt}'(\boxed{\mathsf{sk}^{(i)}, \mathsf{pk}, \mathsf{pk}^*}, C)$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}^{(w^i)}_{\mathsf{BTE}}, \cdot)$, pk as $(\mathsf{pk}_{\mathsf{BTE}}, \cdot)$, and pk^* as $(\cdot, \mathsf{pk}^*_{\mathsf{HPKE}})$. Choose $\varDelta \leftarrow \varPhi^+$, run

$$\boxed{\mathsf{dk}' \leftarrow \mathsf{Enc}_{\mathsf{HPKE}}(\mathsf{pk}^*_{\mathsf{HPKE}}, \mathsf{sk}^{(w^i)}_{\mathsf{BTE}} + \nu^{(w^i)}(\varDelta))} \text{ and}$$

$$\boxed{C' \leftarrow \mathsf{Enc}_{\mathsf{BTE}}(\mathsf{pk} \cdot \mu(\varDelta), \mathsf{Dec}_{\mathsf{BTE}}(\mathsf{sk}^{(i)}, C), i)}. \text{ Return } (\mathsf{dk}', C').$$

Transition$^{0\to1}$: The distributions of Game 0 and Game 1 are indistinguishable under the BTE's adaptability of ciphertexts.

Game 2: We further modify the simulation of Adapt' as follows:

$\mathsf{Adapt}'(\boxed{\mathsf{sk}^{(i)}, \mathsf{pk}^*}, C)$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}^{(w^i)}_{\mathsf{BTE}}, \cdot)$, pk as $(\mathsf{pk}_{\mathsf{BTE}}, \cdot)$, and pk^* as $(\cdot, \mathsf{pk}^*_{\mathsf{HPKE}})$. Choose $\boxed{\mathsf{pk}'_{\mathsf{BTE}}, \mathsf{sk}^{',(\varepsilon)}_{\mathsf{BTE}}, \mathsf{ek}^{',(\varepsilon)}_{\mathsf{BTE}} \leftarrow \mathsf{Gen}_{\mathsf{BTE}}}$ and evolve the secret key to period i, run

$$\boxed{\mathsf{dk}' \leftarrow \mathsf{Enc}_{\mathsf{HPKE}}(\mathsf{pk}^*_{\mathsf{HPKE}}, \mathsf{sk}^{\prime,(w^i)}_{\mathsf{BTE}})} \text{ and}$$

$$\boxed{C' \leftarrow \mathsf{Enc}_{\mathsf{BTE}}(\mathsf{pk}'_{\mathsf{BTE}}, \mathsf{Dec}_{\mathsf{BTE}}(\mathsf{sk}^{(i)}, C), i)}. \text{ Return } (\mathsf{dk}', C').$$

Transition$^{1\to 2}$: The change is conceptual.

In Game 2 all the secret BTE keys the adversary gets are chosen independently from the challenge key. Hence, Game 2 is a standard IND-CPA game and thus the success probability of Game 2 is negligible by Theorem 1. □

Now, given an adaptable fs-DPKE scheme, we use the Adapt algorithm to obtain a fs-PRE$^+$ secure fs-PRE scheme. While the algorithms Setup, Gen, Evo, Enc$^{(i)}$, and Dec$^{(i)}$ can simply be lifted from the fs-DPKE scheme, we note that for each period j in the fs-PRE scheme, we use two periods, i.e., $2j-1$ and $2j$, of the fs-DPKE scheme. The period $2j-1$ is used for level 1 ciphertexts whereas the period $2j$ is used for level 2 ciphertexts[15]. We use $\mathsf{Del}_{\mathsf{DPKE}}$ and $\mathsf{DelEvo}_{\mathsf{DPKE}}$ for ReGen and ReEvo, respectively. For the re-encryption algorithm ReEnc, we apply Adapt. Dec$^{(1)}$ for re-encrypted ciphertexts then decrypts the ciphertext by running $\mathsf{DelDec}_{\mathsf{DPKE}}$ on the adapted delegation key and ciphertext. The full scheme is presented in Scheme 3.

We prove that our scheme is both fs-IND-CPA-1 and fs-IND-CPA-2 secure. Both security notions follow from the fs-IND-CPA security of the underlying fs-DPKE scheme. In contrast, to achieve fs-RIND-CPA, we require an fs-ADAP-IND-CPA fs-DPKE scheme.

Theorem 3. *If instantiated with a fs-IND-CPA and fs-ADAP-IND-CPA secure fs-*DPKE *scheme, Scheme 3 is a fs-*PRE$^+$*-secure fs-*PRE *scheme.*

Proof. Informally speaking, the security experiment for fs-IND-CPA-2 with a fixed period j^* corresponds to the fs-IND-CPA experiment for fs-DPKE for period $2j^*$. We can build a straightforward reduction from an adversary against fs-IND-CPA-2, A_2 to fs-IND-CPA for fs-DPKE:

- When started on pp, n and pk, run $(j^*, \mathsf{st}) \leftarrow A_2(\mathsf{pp}, \lceil \frac{n}{2} + 1 \rceil, \mathsf{pk})$. Set $j' \leftarrow 2j^*$ and return (j', st).
- When started on $\mathsf{st}, \mathsf{sk}^{(j')}_{\mathsf{DPKE}}, \mathsf{ek}^{(j')}_{\mathsf{DPKE}}$, we simulate the $\mathsf{ReGen}^{(h)}$ and $\mathsf{ReGen}^{(h')}$ oracles using $\mathsf{Del}^{(h)}$ and $\mathsf{Del}^{(h')}$. Indeed, $\mathsf{Del}^{(h)}$ and $\mathsf{Del}^{(h')}$ return delegation keys for period $j'-1 = 2j^*-1$, which are re-encryption keys for period j^*-1. Using Evo we evolve $\mathsf{sk}^{(j')}_{\mathsf{DPKE}}, \mathsf{ek}^{(j')}_{\mathsf{DPKE}}$ to period $j'+1$. Set $(\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)}) \leftarrow ((\mathsf{sk}^{(j')}_{\mathsf{DPKE}}, \mathsf{sk}^{(j'+1)}_{\mathsf{DPKE}}), (\mathsf{ek}^{(j')}_{\mathsf{DPKE}}, \mathsf{ek}^{(j'+1)}_{\mathsf{DPKE}}))$ and start A_2 on $\mathsf{st}, (\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$ and simply forward the result.
- Finally, when started on st and $C_{j'-1}$, $C_{j'-1}$ is a level 2 ciphertext for j^*-1. Hence we start A_2 on the ciphertext and return its' result.

[15] One can see the keys for period $2j$ as weak keys in the sense of [4, Third Attempt] whereas the keys for period $2j-1$ constitute the master secret keys.

Let $(\mathsf{Setup}_{\mathsf{DPKE}}, \mathsf{Gen}_{\mathsf{DPKE}}, \mathsf{Evo}_{\mathsf{DPKE}}, \mathsf{Del}_{\mathsf{DPKE}}, \mathsf{Enc}_{\mathsf{DPKE}}, \mathsf{Dec}_{\mathsf{DPKE}}, \mathsf{Adapt}_{\mathsf{DPKE}})$ be fs-DPKE scheme with adaption of ciphertexts and delegation keys.

$\underline{\mathsf{Setup}(1^k)}$: Return $\mathsf{Setup}_{\mathsf{DPKE}}(1^k)$.

$\underline{\mathsf{Gen}(\mathsf{pp}, n)}$: Set $(\mathsf{pk}_{\mathsf{DPKE}}, \mathsf{sk}^{(0)}_{\mathsf{DPKE}}, \mathsf{ek}^{(0)}_{\mathsf{DPKE}}) \leftarrow \mathsf{Gen}_{\mathsf{DPKE}}(\mathsf{pp}, 2n{+}1)$, obtain $(\mathsf{sk}^{(1)}_{\mathsf{DPKE}}, \mathsf{ek}^{(1)}_{\mathsf{DPKE}}) \leftarrow \mathsf{Evo}_{\mathsf{DPKE}}(\mathsf{sk}^{(0)}_{\mathsf{DPKE}}, \mathsf{ek}^{(0)}_{\mathsf{DPKE}})$, and return $(\mathsf{pk}_{\mathsf{DPKE}}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)})$, where

$$\mathsf{sk}^{(0)} \leftarrow (\mathsf{sk}^{(0)}_{\mathsf{DPKE}}, \mathsf{sk}^{(1)}_{\mathsf{DPKE}}),\ \mathsf{ek}^{(0)} \leftarrow (\mathsf{ek}^{(0)}_{\mathsf{DPKE}}, \mathsf{ek}^{(1)}_{\mathsf{DPKE}}).$$

$\underline{\mathsf{Evo}(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})}$: Parse $(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})$ as $((\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}}), (\mathsf{ek}^{(2i)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+1)}_{\mathsf{DPKE}}))$ and return $(\mathsf{sk}^{(i+1)}, \mathsf{ek}^{(i+1)}) = (\mathsf{sk}^{(2i+2)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+3)}_{\mathsf{DPKE}}), (\mathsf{ek}^{(2i+2)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+3)}_{\mathsf{DPKE}}))$, where

$$(\mathsf{sk}^{(2i+1+j)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+1+j)}_{\mathsf{DPKE}}) \leftarrow \mathsf{Evo}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i+j)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+j)}_{\mathsf{DPKE}}) \text{ for } j \in [2].$$

$\underline{\mathsf{Enc}^{(1)}(\mathsf{pk}, M, i)}$: Return $\mathsf{Enc}_{\mathsf{DPKE}}(\mathsf{pk}, M, 2i)$.

$\underline{\mathsf{Enc}^{(2)}(\mathsf{pk}, M, i)}$: Return $\mathsf{Enc}_{\mathsf{DPKE}}(\mathsf{pk}, M, 2i + 1)$.

$\underline{\mathsf{Dec}^{(1)}(\mathsf{sk}^{(i)}, C)}$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}})$ and return $\mathsf{Dec}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i)}, C)$ if C was not re-encrypted. Otherwise parse C as (C_1, rk) and return $\mathsf{DelDec}_{\mathsf{DPKE}}($ $\mathsf{sk}^{(2i+1)}, \mathsf{rk}, C_1)$.

$\underline{\mathsf{Dec}^{(2)}(\mathsf{sk}^{(i)}, C)}$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}})$ and return $\mathsf{Dec}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i+1)}, C)$.

$\underline{\mathsf{ReGen}(\mathsf{sk}^{(i)}_A, \mathsf{ek}^{(i)}_A, \mathsf{pk}_B)}$: Parse $(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})$ as $((\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}}), (\mathsf{ek}^{(2i)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+1)}_{\mathsf{DPKE}}))$, and $\mathsf{Del}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i+1)}_A, \mathsf{ek}^{(2i+1)}_A, \mathsf{pk}_B)$.

$\underline{\mathsf{ReEvo}(\mathsf{rk}^{(i)}_{A\to B}, \mathsf{rek}^{(i)}_{A\to B})}$: Return $\mathsf{DelEvo}_{\mathsf{DPKE}}(\mathsf{DelEvo}_{\mathsf{DPKE}}(\mathsf{rk}^{(i)}_{A\to B}, \mathsf{rek}^{(i)}_{A\to B}))$.

$\underline{\mathsf{ReEnc}(\mathsf{rk}^{(i)}_{A\to B}, C_A)}$: Choose $\tau \xleftarrow{R} \mathbb{G}$ and return $\mathsf{Adapt}_{\mathsf{DPKE}}(\mathsf{rk}^{(i)}_{A\to B}, C_A, \tau)$.

Scheme 3. fs-PRE scheme from an adaptable fs-DPKE scheme.

To show fs-IND-CPA-1 security, we perform a similar reduction:

- When started on pp, n and pk, run $(j^*, \mathsf{st}) \leftarrow A_1(\mathsf{pp}, \lceil \frac{n}{2} + 1 \rceil, \mathsf{pk})$. Set $j' \leftarrow 2j^* - 1$ and return (j', st).
- When started on $\mathsf{st}, \mathsf{sk}^{(j')}_{\mathsf{DPKE}}, \mathsf{ek}^{(j')}_{\mathsf{DPKE}}$, we simulate the $\mathsf{ReGen}^{(h)}$ and $\mathsf{ReGen}^{(h')}$ oracles using $\mathsf{Del}^{(h)}$ and $\mathsf{Del}^{(h')}$ and by running DelEvo on the result. Indeed, $\mathsf{Del}^{(h)}$ and $\mathsf{Del}^{(h')}$ return delegation keys for period $j' - 1 = 2j^* - 2$, hence after applying DelEvo we obtain re-encryption keys for period $j^* - 1$. $\mathsf{ReGen}^{(d)}$ is simulated honestly by delegating $\mathsf{sk}^{(j')}_{\mathsf{DPKE}}, \mathsf{ek}^{(j')}_{\mathsf{DPKE}}$ to a dishonest user. Using Evo we evolve $\mathsf{sk}^{(j')}_{\mathsf{DPKE}}, \mathsf{ek}^{(j')}_{\mathsf{DPKE}}$ to period $j' + 2$. Set $(\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)}) \leftarrow ((\mathsf{sk}^{(j'+1)}_{\mathsf{DPKE}}, \mathsf{sk}^{(j'+2)}_{\mathsf{DPKE}}), (\mathsf{ek}^{(j'+1)}_{\mathsf{DPKE}}, \mathsf{ek}^{(j'+2)}_{\mathsf{DPKE}}))$ and start A_1 on $\mathsf{st}, (\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$ and simply forward the result.
- Finally, when started on st and $C_{j'-1}$, $C_{j'-1}$ is a level 1 ciphertext for $j^* - 1$. Hence we start A_1 on the ciphertext and return its' result.

To show receiver-IND-CPA security we build an fs-ADAP-IND-CPA adversary against the fs-DPKE scheme. The fs-RIND-CPA adversary is denoted as A_r.

- When started on pp, n and pk, run $(j^*, \mathsf{st}) \leftarrow A_r(\mathsf{pp}, \lceil \frac{n}{2} + 1 \rceil, \mathsf{pk})$. Set $j' \leftarrow 2j^* + 1$ and return (j', st).

- When started on st, we can simulate ReEnc honestly using Adapt.
- Wen started on st and C, the ciphertext is a level 2 ciphertext for period $j*$, hence we return $A_r(\mathsf{st}, C)$.

Note that all values are consistently distributed in all three reductions. □

4.4 Separating fs-PRE^- from fs-PRE^+

To expand on the gap between fs-PRE^+ and fs-PRE^- schemes and to provide an explicit separation, we construct a counterexample. In particular, it is clear that every scheme that satisfies fs-PRE^+ also satisfies fs-PRE^-. For our separation we now present a scheme that is fs-PRE^- but trivially violates fs-PRE^+. The scheme is also built from a fs-DPKE scheme and presented in Scheme 4. In this scheme however, ReEnc simply embeds the delegation key in the re-encrypted ciphertext. The shortcomings of this construction compared to Scheme 3 are obvious: once the receiver is presented with one valid re-encrypted ciphertext, it can recover the delegation key from that ciphertext and can decrypt all level 2 ciphertexts for this period.

Let $(\mathsf{Setup}_{\mathsf{DPKE}}, \mathsf{Gen}_{\mathsf{DPKE}}, \mathsf{Evo}_{\mathsf{DPKE}}, \mathsf{Del}_{\mathsf{DPKE}}, \mathsf{Enc}_{\mathsf{DPKE}}, \mathsf{Dec}_{\mathsf{DPKE}})$ be fs-DPKE scheme.

$\underline{\mathsf{Setup}(1^k)}$: Return $\mathsf{Setup}_{\mathsf{DPKE}}(1^k)$.

$\underline{\mathsf{Gen}(\mathsf{pp}, n)}$: Set $(\mathsf{pk}_{\mathsf{DPKE}}, \mathsf{sk}^{(0)}_{\mathsf{DPKE}}, \mathsf{ek}^{(0)}_{\mathsf{DPKE}}) \leftarrow \mathsf{Gen}_{\mathsf{DPKE}}(\mathsf{pp}, 2n+1)$, obtain $(\mathsf{sk}^{(1)}_{\mathsf{DPKE}}, \mathsf{ek}^{(1)}_{\mathsf{DPKE}}) \leftarrow \mathsf{Evo}_{\mathsf{DPKE}}(\mathsf{sk}^{(0)}_{\mathsf{DPKE}}, \mathsf{ek}^{(0)}_{\mathsf{DPKE}})$, and return $(\mathsf{pk}_{\mathsf{DPKE}}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)})$, where

$$\mathsf{sk}^{(0)} \leftarrow (\mathsf{sk}^{(0)}_{\mathsf{DPKE}}, \mathsf{sk}^{(1)}_{\mathsf{DPKE}}),\ \mathsf{ek}^{(0)} \leftarrow (\mathsf{ek}^{(0)}_{\mathsf{DPKE}}, \mathsf{ek}^{(1)}_{\mathsf{DPKE}}).$$

$\underline{\mathsf{Evo}(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})}$: Parse $(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})$ as $((\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}}), (\mathsf{ek}^{(2i)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+1)}_{\mathsf{DPKE}}))$ and return $(\mathsf{sk}^{(i+1)}, \mathsf{ek}^{(i+1)}) = (\mathsf{sk}^{(2i+2)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+3)}_{\mathsf{DPKE}}), (\mathsf{ek}^{(2i+2)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+3)}_{\mathsf{DPKE}}))$, where

$$(\mathsf{sk}^{(2i+1+j)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+1+j)}_{\mathsf{DPKE}}) \leftarrow \mathsf{Evo}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i+j)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+j)}_{\mathsf{DPKE}}) \text{ for } j \in [2].$$

$\underline{\mathsf{Enc}^{(1)}(\mathsf{pk}, M, i)}$: Return $\mathsf{Enc}_{\mathsf{DPKE}}(\mathsf{pk}, M, 2i)$.

$\underline{\mathsf{Enc}^{(2)}(\mathsf{pk}, M, i)}$: Return $\mathsf{Enc}_{\mathsf{DPKE}}(\mathsf{pk}, M, 2i+1)$.

$\underline{\mathsf{Dec}^{(1)}(\mathsf{sk}^{(i)}, C)}$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}})$ and return $\mathsf{Dec}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i)}, C)$ if C was not re-encrypted. Otherwise parse C as (C_1, rk) and return $\mathsf{DelDec}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i+1)}, \mathsf{rk}, C_1)$.

$\underline{\mathsf{Dec}^{(2)}(\mathsf{sk}^{(i)}, C)}$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}})$ and return $\mathsf{Dec}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i+1)}, C)$.

$\underline{\mathsf{ReGen}(\mathsf{sk}^{(i)}_A, \mathsf{ek}^{(i)}_A, \mathsf{pk}_B)}$: Parse $(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})$ as $((\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}}), (\mathsf{ek}^{(2i)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+1)}_{\mathsf{DPKE}}))$, and return $(\mathsf{rk}^{(i)}_{A\to B}, \mathsf{rek}^{(i)}_{A\to B})$, where

$$(\mathsf{rk}^{(i)}_{A\to B}, \mathsf{rek}^{(i)}_{A\to B}) \leftarrow \mathsf{Del}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i+1)}_A, \mathsf{ek}^{(2i+1)}_A, \mathsf{pk}_B).$$

$\underline{\mathsf{ReEvo}(\mathsf{rk}^{(i)}_{A\to B}, \mathsf{rek}^{(i)}_{A\to B})}$: Return $\mathsf{DelEvo}_{\mathsf{DPKE}}(\mathsf{DelEvo}_{\mathsf{DPKE}}(\mathsf{rk}^{(i)}_{A\to B}, \mathsf{rek}^{(i)}_{A\to B}))$.

$\underline{\mathsf{ReEnc}(\mathsf{rk}^{(i)}_{A\to B}, C_A)}$: Return $(C_A, \mathsf{rk}^{(i)}_{A\to B})$.

Scheme 4. fs-PRE scheme from a fs-DPKE scheme without adaption.

In the following Theorem, we first show that Scheme 4 is indeed fs-PRE^- secure, i.e., satisfies fs-IND-CPA-1 and fs-IND-CPA-2 security, but trivially does not satisfy fs-RIND-CPA security and thus is not fs-PRE^+ secure.

Theorem 4. *Scheme 4 when instantiated with a fs-IND-CPA secure fs-*DPKE *scheme satisfies fs-IND-CPA-1 and fs-IND-CPA-2 security, but not fs-RIND-CPA security.*

Proof. We follow the same strategy as for Theorem 3 to show fs-IND-CPA-2.

- When started on pp, n and pk, run $(j^*, \mathsf{st}) \leftarrow A_2(\mathsf{pp}, \lceil \frac{n}{2} + 1 \rceil, \mathsf{pk})$. Set $j' \leftarrow 2j^*$ and return (j', st).
- When started on $\mathsf{st}, \mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j')}$, we simulate the ReGen^h and $\mathsf{ReGen}^{h'}$ oracles using Del^h and $\mathsf{Del}^{h'}$. Indeed, Del^h and $\mathsf{Del}^{h'}$ return delegation keys for period $j' - 1 = 2j^* - 1$, which are re-encryption keys for period $j^* - 1$. Using Evo we evolve $\mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j')}$ to period $j' + 1$. Set $(\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)}) \leftarrow ((\mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{sk}_{\mathsf{DPKE}}^{(j'+1)}), (\mathsf{ek}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j'+1)}))$ and start A_2 on $\mathsf{st}, (\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$ and simply forward the result.
- Finally, when started on st and $C_{j'-1}$, $C_{j'-1}$ is a level 2 ciphertext for $j^* - 1$. Hence we start A_2 on the ciphertext and return its' result.

To show fs-IND-CPA-1 security, we perform a similar reduction:

- When started on pp, n and pk, run $(j^*, \mathsf{st}) \leftarrow A_1(\mathsf{pp}, \lceil \frac{n}{2} + 1 \rceil, \mathsf{pk})$. Set $j' \leftarrow 2j^* - 1$ and return (j', st).
- When started on $\mathsf{st}, \mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j')}$, we simulate the ReGen^h and $\mathsf{ReGen}^{h'}$ oracles using Del^h and $\mathsf{Del}^{h'}$ and by running DelEvo on the result. Indeed, Del^h and $\mathsf{Del}^{h'}$ return delegation keys for period $j' - 1 = 2j^* - 2$, hence after applying DelEvo we obtain re-encryption keys for period $j^* - 1$. ReGen^d is simulated honestly by delegating $\mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j')}$ to a dishonest user. Using Evo we evolve $\mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j')}$ to period $j' + 2$. Set $(\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)}) \leftarrow ((\mathsf{sk}_{\mathsf{DPKE}}^{(j'+1)}, \mathsf{sk}_{\mathsf{DPKE}}^{(j'+2)}), (\mathsf{ek}_{\mathsf{DPKE}}^{(j'+1)}, \mathsf{ek}_{\mathsf{DPKE}}^{(j'+2)}))$ and start A_1 on $\mathsf{st}, (\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$ and simply forward the result.
- Finally, when started on st and $C_{j'-1}$, $C_{j'-1}$ is a level 1 ciphertext for $j^* - 1$. Hence we start A_1 on the ciphertext and return its' result.

Following the initial observation on the recoverability of delegation keys, an receiver-IND-CPA adversary is straightforward to define:

- When started on pp, n and pk, honestly generate a key $(\mathsf{pk}^*, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n)$ and store it in st. Choose $j^* \xleftarrow{R} [n]$ and store it together with pk in st, and return $(j^*, \mathsf{pk}^*, \mathsf{st})$.
- When started on st to output the challenge messages, choose $M_0, M_1, M_2 \xleftarrow{R} \mathcal{M}$. Invoke the ReEnc oracle as $(\cdot, \mathsf{dk}) \leftarrow \mathsf{ReEnc}(\mathsf{rk}, \mathsf{Enc}^{(2)}(\mathsf{pk}, M_2, j^*))$ and store M_0, M_1, dk in st. Return M_0, M_1, st.

- Now when started on st and the challenge ciphertext C, use dk stored in st and obtain $M \leftarrow \mathsf{DelDec}_{\mathsf{DPKE}}(\mathsf{sk}^{(2j^*+1)}, \mathsf{dk}, C)$. Check for which $i \in \{0,1\}$ $M = M_i$ and return i.

Regardless of the chosen period the adversary always wins, rendering the scheme insecure with respect to the fs-RIND-CPA notion. □

From this theorem we obtain the following corollary:

Corollary 1. *fs-PRE$^+$ is a strictly stronger notion than fs-PRE$^-$.*

Note that this also shows that for conventional PRE scheme there is a separation between the classical security notion of PRE (PRE^-) as defined by Ateniese et al. and the PRE^+ notion.

Acknowledgments. Supported by H2020 project Prismacloud, grant agreement n°644962 and by H2020 project Credential, grant agreement n°653454. We thank all anonymous reviewers for their valuable comments.

References

1. Abdalla, M., An, J.H., Bellare, M., Namprempre, C.: From identification to signatures via the fiat-shamir transform: Minimizing assumptions for security and forward-security. In: EUROCRYPT (2002)
2. Applebaum, B., Harnik, D., Ishai, Y.: Semantic Security under Related-Key Attacks and Applications. In: ICS (2011)
3. Ateniese, G., Benson, K., Hohenberger, S.: Key-private proxy re-encryption. In: CT-RSA (2009)
4. Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage. In: NDSS (2005)
5. Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Trans. Inf. Syst. Secur. 9(1) (2006)
6. Bellare, M., Miner, S.K.: A forward-secure digital signature scheme. In: CRYPTO (1999)
7. Bellare, M., Yee, B.S.: Forward-security in private-key cryptography. In: CT-RSA (2003)
8. Berners-Lee, E.: Improved security notions for proxy re-encryption to enforce access control. IACR Cryptology ePrint Archive 2017, 824 (2017)
9. Blaze, M., Bleumer, G., Strauss, M.: Divertible protocols and atomic proxy cryptography. In: EUROCRYPT (1998)
10. Blazy, O., Bultel, X., Lafourcade, P.: Two secure anonymous proxy-based data storages. In: SECRYPT (2016)
11. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: CRYPTO (2004)
12. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: CRYPTO (2005)
13. Boneh, D., Waters, B.: A fully collusion resistant broadcast, trace, and revoke system. In: CCS (2006)
14. Borceaa, C., Guptaa, A.B.D., Polyakova, Y., Rohloffa, K., Ryana, G.: Picador: End-to-end encrypted publish-subscribe information distribution with proxy re-encryption. Future Generation Comp. Syst. (2016)

15. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: EUROCRYPT (2003)
16. Canetti, R., Hohenberger, S.: Chosen-ciphertext secure proxy re-encryption. In: CCS (2007)
17. Canetti, R., Raghuraman, S., Richelson, S., Vaikuntanathan, V.: Chosen-ciphertext secure fully homomorphic encryption. In: PKC 2017 (2017)
18. Chandran, N., Chase, M., Liu, F., Nishimaki, R., Xagawa, K.: Re-encryption, functional re-encryption, and multi-hop re-encryption: A framework for achieving obfuscation-based security and instantiations from lattices. In: PKC (2014)
19. Chandran, N., Chase, M., Vaikuntanathan, V.: Functional re-encryption and collusion-resistant obfuscation. In: TCC (2012)
20. Cohen, A.: What about bob? the inadequacy of CPA security for proxy reencryption. IACR Cryptology ePrint Archive 2017, 785 (2017)
21. Cohen, A., Holmgren, J., Nishimaki, R., Vaikuntanathan, V., Wichs, D.: Watermarking cryptographic capabilities. In: STOC (2016)
22. Delerablée, C.: Identity-based broadcast encryption with constant size ciphertexts and private keys. In: ASIACRYPT 2007 (2007)
23. Fan, X., Liu, F.: Proxy re-encryption and re-signatures from lattices. IACR Cryptology ePrint Archive 2017, 456 (2017)
24. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC (2009)
25. Gentry, C., Silverberg, A.: Hierarchical id-based cryptography. In: ASIACRYPT (2002)
26. Green, M., Ateniese, G.: Identity-based proxy re-encryption. In: ACNS (2007)
27. Green, M.D., Miers, I.: Forward secure asynchronous messaging from puncturable encryption. In: IEEE S&P (2015)
28. Günther, C.G.: An identity-based key-exchange protocol. In: EUROCRYPT (1989)
29. Günther, F., Hale, B., Jager, T., Lauer, S.: 0-rtt key exchange with full forward secrecy. In: EUROCRYPT (2017)
30. Hanaoka, G., Kawai, Y., Kunihiro, N., Matsuda, T., Weng, J., Zhang, R., Zhao, Y.: Generic construction of chosen ciphertext secure proxy re-encryption. In: CT-RSA (2012)
31. Hohenberger, S., Rothblum, G.N., Shelat, A., Vaikuntanathan, V.: Securely obfuscating re-encryption. J. Cryptology (2011)
32. Libert, B., Vergnaud, D.: Tracing malicious proxies in proxy re-encryption. In: Pairing (2008)
33. Libert, B., Vergnaud, D.: Unidirectional chosen-ciphertext secure proxy re-encryption. In: PKC (2008)
34. Libert, B., Vergnaud, D.: Unidirectional chosen-ciphertext secure proxy re-encryption. IEEE Trans. Information Theory (2011)
35. Myers, S., Shull, A.: Efficient hybrid proxy re-encryption for practical revocation and key rotation. IACR Cryptology ePrint Archive 2017, 833 (2017)
36. Polyakov, Y., Rohloff, K., Sahu, G., Vaikuntanathan, V.: Fast proxy re-encryption for publish/subscribe systems. ACM Trans. Priv. Secur. 20(4), 14:1–14:31 (2017)
37. Ren, Y., Gu, D., Wang, S., Zhang, X.: Hierarchical identity-based proxy re-encryption without random oracles. Int. J. Found. Comput. Sci. 21(6), 1049–1063 (2010)
38. Sakai, R., Furukawa, J.: Identity-based broadcast encryption. IACR Cryptology ePrint Archive (2007)
39. Tang, Q.: Type-based proxy re-encryption and its construction. In: INDOCRYPT (2008)

40. Tessaro, S., Wilson, D.A.: Bounded-collusion identity-based encryption from semantically-secure public-key encryption: Generic constructions with short ciphertexts. In: PKC (2014)
41. Weng, J., Yang, Y., Tang, Q., Deng, R.H., Bao, F.: Efficient conditional proxy re-encryption with chosen-ciphertext security. In: ISC 2009 (2009)
42. Xu, P., Xu, J., Wang, W., Jin, H., Susilo, W., Zou, D.: Generally hybrid proxy re-encryption: A secure data sharing among cryptographic clouds. In: AsiaCCS (2016)

4

Short Double- and N-Times-Authentication-Preventing Signatures from ECDSA and More

Publication Data

David Derler, Sebastian Ramacher, and Daniel Slamanig. Short Double- and N-Times-Authentication-Preventing Signatures from ECDSA and More. In *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018*, pages 273–287. IEEE, 2018. URL: https://doi.org/10.1109/EuroSP.2018.00027

The appended paper is an author-created full version available at at https://eprint.iacr.org/2017/1203.

Contributions

- The author is one of the main authors.

Short Double- and N-Times-Authentication-Preventing Signatures from ECDSA and More[⋆]

David Derler[1], Sebastian Ramacher[1], and Daniel Slamanig[2]

[1] IAIK, Graz University of Technology, Austria
[2] AIT Austrian Institute of Technology GmbH, Vienna, Austria
firstname.lastname@tugraz.at, firstname.lastname@ait.ac.at

Abstract. Double-authentication-preventing signatures (DAPS) are signatures designed with the aim that signing two messages with an identical first part (called address) but different second parts (called payload) allows to publicly extract the secret signing key from two such signatures. A prime application for DAPS is disincentivizing and/or penalizing the creation of two signatures on different payloads within the same address, such as penalizing double spending of transactions in Bitcoin by the loss of the double spender's money.

So far DAPS have been constructed from very specific signature schemes not used in practice and using existing techniques it has proved elusive to construct DAPS schemes from signatures widely used in practice. This, unfortunately, has prevented practical adoption of this interesting tool so far. In this paper we ask whether one can construct DAPS from signature schemes used in practice. We affirmatively answer this question by presenting novel techniques to generically construct provably secure DAPS from a large class of discrete logarithm based signatures. This class includes schemes like Schnorr, DSA, EdDSA, and, most interestingly for practical applications, the widely used ECDSA signature scheme. The resulting DAPS are highly efficient and the shortest among all existing DAPS schemes. They are nearly half of the size of the most efficient factoring based schemes (IACR PKC'17) and improve by a factor of 100 over the most efficient discrete logarithm based ones (ACM CCS'15). Although this efficiency comes at the cost of a reduced address space, i.e., size of keys linear in the number of addresses, we will show that this is not a limitation in practice. Moreover, we generalize DAPS to any $N > 2$, which we denote as N-times-authentication-preventing signatures (NAPS). Finally, we also provide an integration of our ECDSA-based DAPS into the OpenSSL library and perform an extensive comparison with existing approaches.

[⋆] This is the full version of a paper which appears in 2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018. ©IEEE, 2018.

1 Introduction

Digital signatures are the prevalent cryptographic primitive to provide strong integrity and authenticity guarantees for messages exchanged in the digital realm. They are used in major cryptographic protocols such as TLS, for issuing digital certificates (i.e., certifying public keys) within public-key infrastructures (PKIs), to authenticate executable code or digital documents such as PDF documents (in a legally binding way) or to sign transactions within the distributed cryptocurrency Bitcoin, to name some popular applications. Arguably, as they enable the secure distribution and transmission of public keys, in a very real sense, they serve as the foundation of all public key cryptography in practice.

Most widely used signature schemes today are (1) RSA-FDH, either used with PKCS#1 v1.5 padding or as probabilistic signature scheme (RSA-PSS), and (2) the discrete logarithm based (elliptic curve) digital signature algorithm (EC)DSA. While RSA is predominant in legacy applications, more recent applications that make heavy use of digital signatures (such as Bitcoin) build upon ECDSA. Actually, when analyzing the trend of the use of ECDSA for certificate signing, we can observe that its use is becoming increasingly popular over the last few years[3] (see Table 1). A similar trend can be observed in DNSSEC in that an ever increasing number of DNSSEC resolvers support ECDSA[4] and some large companies like CloudFlare are heavily pushing ECDSA [vRJS16]. Papadopoulos et al. [PWH+17] argue that due to improved performance and security it is very likely that new features for DNSSEC such as NSEC5 will only target the elliptic curve setting instead of RSA. Actually, given that the use of RSA signatures within DNSSEC in practice suffers from deficient key generation methods [SW17], switching to elliptic curves seems to be a viable way to go.

Year	**% of ECDSA signatures**
2014	0.01 %
2015	0.02 %
2016	2.54 %
2017	36.07 %

Table 1: Usage of ECDSA signatures in certificates of the top million websites via `censys.io` [DAM+15].

Now let us recall digital signatures more technically. We have a signer who holds a secret signing key sk and publishes its corresponding public verification key pk. To sign a message m, the signer uses sk to produce a signature σ and anyone who is given (m, σ) together with an authentic copy of pk can verify that the message originated from the signer (authenticity) and has not been modified

[3] https://blog.cloudflare.com/aes-cbc-going-the-way-of-the-dodo/

[4] https://blog.apnic.net/2016/10/06/dnssec-and-ecdsa/

in any way (integrity). Formal security guarantees for a signature scheme require that anyone not holding sk, even if allowed to adaptively obtain signatures for messages of one's choice, will not be able to come up with a valid signature for a non-queried message, i.e., produce a forgery. This notion is coined existential unforgeability under chosen message attacks ($\mathsf{EUF\text{-}CMA}$), formally discussed in Section 4.1, and is the widely accepted security notion required by schemes used in practice today.

In this paper we consider a variant of signature schemes dubbed double-authentication-preventing signatures (DAPS) [PS14,PS17]. Here, messages to be signed are of the form $m = (a, p)$ and in particular they consist of an address a and a payload p. The basic idea behind DAPS is that they behave exactly like conventional signatures, i.e., provide unforgeability in the $\mathsf{EUF\text{-}CMA}$ sense, as long as no distinct payloads $p' \neq p$ are signed with respect to the same address a. If a signer produces two signatures for distinct payloads $p' \neq p$ *but* with respect to the same address a (called colliding messages), then *anyone* can compute the signer's secret key sk from these signatures (the so called double-signature extraction property).

This concept may sound awkward at first sight, but it is indeed interesting as it disincentivizes the signer from "double-signing". It suggests the use of DAPS instead of conventional signatures whenever double-signing should be disincentivized, where the address a (or its associated space respectively) can be given some application-dependent semantics. Thereby, we can consider any form of a digital processes where one wants to prevent fraud by discouraging users from submitting (signing) duplicates. Think for instance of requests for reimbursements for the same expense multiple times, which can be disincentivized when using some unique ID, identifying the invoice/payment as address. In Section 2 we discuss some representative and more concrete applications of DAPS.

We observe that this is conceptually related to some other approaches discussed subsequently, but DAPS are stronger in the sense that they reveal the secret key of the signer to the public. Within offline double spending mechanisms [CFN88] of centralized e-cash systems, as long as a user is honest, the user can anonymously conduct transactions. But if a user misbehaves and spends an e-coin multiple times, his identity is revealed. In contrast to just revealing the identity in case of misbehaviour, however, DAPS reveal the secret key of the signer. Revealing the secret key as discouragement to behave fraudulent is also related to what is done within the so called PKI-assured non-transferability approach in anonymous credential systems [CL01]. Here the secret of the credential is associated to a valuable secret outside the system, e.g., a secret key that allows to issue signatures that are equivalent to handwritten signatures, which disincentivizes the sharing of a credential. However, in contrast to DAPS the secret key is not made public per se, but known to everyone with whom the credential is shared.

A problem with existing DAPS constructions [PS14,RKS15,PS17,BPS17] is that they are not based on widely used signature schemes and thus have not seen adoption in practice. While the constructions in [PS14,PS17,BPS17] are factoring based ones (aka in the RSA setting), the one from Ruffing et al. in [RKS15] is compatible with discrete-logarithm based signature public keys (and ECDSA public keys in particular). Unfortunately, their integration of signature public keys in so called accountable assertions[5], which Ruffing et al. instantiate with a Merkle-tree construction using chameleon hash functions [KR00], does not yield an efficient construction. Our aim in this paper is to provide a generic construction that augments existing signature schemes widely used in practice (such as ECDSA) to yield DAPS being provably secure, where the security proof makes only black-box use of the signature scheme.

1.1 Contribution

Our key contributions in this paper can be summarized as follows:

- We are the first to present DAPS that are based on widely deployed and used signature schemes and in particular ECDSA. Additionally, our approach also works identically for Schnorr signatures, DSA or EdDSA (and many other discrete-logarithm based schemes). Consequently, we provide the first construction that can be directly used in real world and deployed systems.
- We introduce notions of double-signing extraction security for DAPS schemes that extend keys of a conventional signature scheme. Our notions ensure that extractability of the signing key of the signature scheme, e.g., the ECDSA key, is required, even if it is not possible to extract the full DAPS secret key. In applications where the signing key is also used in a different context, inadvertently leaking the signing key already disincentivizes double-authentication. We show that our construction satisfies this notion under adversarially chosen, i.e., malicious, keys.
- Our DAPS are the *shortest* DAPS so far in any setting. For instance, for the 128 bit security level, signatures of our DAPS with ECDSA on 256 bit elliptic curve groups are 1280 bits long, whereas most efficient factoring-based DAPS with a modulus size of 2048 bit require 2049 bits. This compactness, however, comes at the cost of a reduced address space and public key size linearly depending on the address space. However, as we will show, practical use-cases only require small address spaces and thus keep the key sizes reasonably low.
- Our construction paradigm is a generic and novel approach to combine verifiable Shamir secret sharing with (linear) ElGamal encryption in a semi-black box way. In a nutshell, the idea is to homomorphically evaluate the verification relation of the verifiable secret sharing scheme in the encrypted domain and to prove that the respective encrypted evaluation actually contains the expected value. This, in turn, gives us the required flexibility to perform a black-box reduction to the $\mathsf{EUF\text{-}CMA}$ security of ECDSA, or, more generally, to the $\mathsf{EUF\text{-}CMA}$ security of any discrete logarithm based signature scheme where the public key is the image of the secret key under a group homomorphism. From a practical point of view, this allows an easy extension of existing (EC)DSA, EdDSA and Schnorr signing keys to DAPS keys.

[5] Ruffing et al. show that certain accountable assertions (and in particular their construction) yield DAPS.

– We generalize DAPS and show how our approach to construct DAPS can easily be extended to N-times-authentication-preventing signatures (dubbed NAPS) for any $N > 2$. This is achieved by setting the degree of the polynomial in Shamir's secret sharing to $N - 1$ (where we simply have a degree 1 polynomial in case of DAPS).
– We provide an implementation of our DAPS and integration into the popular OpenSSL library, which requires no changes to OpenSSL's ECDSA interface and implementation. This allows faster adoption of our DAPS in existing applications such as Bitcoin.

Follow up work. Bertram Poettering made us aware of follow up work on short DAPS in the discrete logarithm setting which appears at Africacrypt 2018 [Poe18]. His DAPS provide noticeably smaller key and signature sizes, extractability of the whole DAPS key, but his work does not allow to extend signature schemes to DAPS in a black box way. In contrast, our results allow to extend signature schemes to DAPS in a black box way, while the extraction notion only allows to extract the key of the signature scheme. Additionally, the work in [Poe18] does not yield NAPS.

2 Applications of DAPS

Below we discuss three appealing applications of DAPS. The first two are applications already given in [RKS15], which can be implemented with our construction much more efficiently. The last field of application is more generic and includes disincentivizing double-signing of certificates and executables.

Moreover, we stress that as our DAPS constructions are the first that are ready to be used based on a widely deployed signature scheme that is used in many real world applications and whose popularity is ever increasing. Thus, we are convinced that DAPS will find many more interesting applications.

2.1 Accountable Assertions and Non-equivocation Contracts

Accountable assertions introduced in [RKS15] are a cryptographic mechanism that allows binding of statements to contexts in an accountable way: if the attacker asserts two contradicting statements in the same context, then any observer can extract the attacker's secret key. DAPS can be viewed as a stronger variant of accountable assertions, as they are additionally required to be unforgeable. Hence efficient DAPS constructions also provide more efficient instantiations of accountable assertions.

Combining accountable assertions respectively DAPS with Bitcoin deposits as discussed in [RKS15] enables the construction of non-equivocation contracts. Latter make it possible to penalize equivocation in distributed protocols monetarily. If a party A should be penalized if it equivocates, A creates a new Bitcoin key pair and extends it to a DAPS key pair.[6] It creates a deposit under the

[6] Ruffing et al. use the signature public key as a public key of a accountable assertion instead of using a DAPS directly.

newly created Bitcoin key pair. Whenever A is supposed to send a statement in some context, it additionally sends a signature under the corresponding DAPS key. If A equivocates, anyone can extract the secret key from the two assertions with respect to the same context and can hence transfer the funds stored in the deposit to an address under their control. In case that A does not equivocate, it keeps full control over the deposit.

2.2 Disincentivizing Bitcoin Double-Spending

A central issue in the Bitcoin protocol is that it takes some time (in the order of tens of minutes) until a transaction gets confirmed in the blockchain and thus becomes valid. This makes it hard to prevent double-spending for "fast" transactions, i.e., transactions which involve transferring goods immediately after completing a transaction. To this end various non-cryptographic means to detect double-spending in fast Bitcoin transactions were proposed [KAC12,KAR$^+$15].

With DAPS we can come up with a cryptographic solution towards solving this problem that strongly disincentivizes double-spending of the aforementioned type. In particular, we can ensure that double-spending will reveal the signing key and thus the associated Bitcoin(s) of the misbehaving party. To achieve this we can follow a similar strategy as [RKS15], but building upon our DAPS yields a much more efficient solution which is suited to be directly added to the Bitcoin core with a few lines of code, i.e., by extending the existing use of ECDSA for signing to our DAPS based on ECDSA. To disincentivize double-spending for a limited number of offline transactions, a user A of a service B first transfers an amount of spendable coins and a penalty to a deposit. After the deposit was confirmed by the blockchain, A can buy services from B offline by signing transactions with the DAPS scheme and giving the signatures to B. Now, if A is honest throughout all transactions, A can clear the deposit after some threshold. However, when A double-spends the DAPS signatures leak the secret (ECDSA) key to B. Thus A looses the coins deposited as penalty, since B is now able to transfer the coins to a wallet under its control.

2.3 Disincentivizing Double-Signing

More generally, DAPS are useful to disincentivize double-signing. Poettering and Stebila [PS14,PS17] propose the use of DAPS for certificate signing within public key infrastructures (PKIs). For this application, it seems that [PS17] is favorable to what we will present. Nevertheless, there are other similar application, where—likewise to the other applications presented in this section—our novel constructions are favorable to prior work.

Think of the application of DAPS in context of code-signing, i.e., for the signing of executables. When DAPS are used, the address represents a unique ID (such as used by Apple's App Store or Google's Play Store) and the payload is the version number. Providing a clean and a backdoored variant of the same software version will leak the signing key. This disincentivizes such a behaviour as this will then likely lead to a pandemia of malware signed with such a key.

2.4 Observation Regarding the Address Space

Interestingly, we observe that none of the applications requires an exponentially large address space. For example the application to accountable assertions inherently only requires a single address. Furthermore, in the application to disincentivizing double-spending for fast Bitcoins transaction, one may observe that a small number of addresses suffices. Consider for example a public transport company that allows customers to charge a transport pass for multiple trips. In this case the number of taken trips can serve as address. Finally, in the application to code signing one requires a somewhat larger address space, but still having an address space of size 100 would allow to sign a new software version every week for about two years.

3 Overview

In the following we provide an overview of the path we take in this paper to construct DAPS. Previous approaches to construct DAPS follow the idea of finding and formalizing some suitable cryptographic primitive that directly allows to obtain DAPS. Examples are 2:1 trapdoor functions as in [PS14,PS17], or certain trapdoor identification schemes as in [BPS17]. While such an approach is highly challenging and interesting from a theoretical perspective, following this approach makes it very unlikely that one ends up with DAPS that are based on some already deployed signature scheme like (EC)DSA. Our approach in this paper is diametrically opposed to this approach. Namely, we look at signature schemes used in practice and ask if and how we can turn them into DAPS. Thereby, we put our focus on the elliptic-curve (discrete logarithm) setting.

The dead end. Before we present our approach we briefly discuss why a seemingly rather obvious path unfortunately does not work, as we consider this finding an interesting observation. When looking at schemes from the ElGamal family [Gam84,HPM94], like (EC)DSA or Schnorr [Sch89] signatures, it is well known that wrong usage may inadvertently leak the entire secret signing key. More precisely, due to the nature of these schemes, using the same randomness for computing signatures on different messages—as already happened in the past either due to erroneously fixing the randomness[7] or due to a bad randomness generation[8]—reveals the secret signing key. While there are countermeasures to avoid the aforementioned issues in practice at all by either making (EC)DSA deterministic [Por13] or by explicitly designing deterministic schemes such as EdDSA [BDL$^+$12], the randomized versions, which are susceptible to the above problem, are still those most commonly used.

Now, one could try to make this aforementioned "bug" a "feature" and use this inherent property of such signature schemes in a positive way to construct DAPS. Recall, that DAPS require extraction of the signing key when given two

[7] http://www.bbc.com/news/technology-12116051

[8] http://www.theregister.co.uk/2013/08/12/android_bug_batters_bitcoin_wallets/

signatures for colliding messages. Now what we could do is to adopt the idea as used by [Por13,BDL+12]. The idea would be to pseudorandomly compute the randomness used for signing from the message and the (secret) key. In contrast to making conventional signatures deterministic, in DAPS we cannot trust the signer to actually compute the randomness pseudorandomly from the address and there must be some means for anyone to check that the signer indeed honestly computed the randomness from the address. Now, one could think that it would work to use a verifiable random function (VRFs) [MRV99] to derive the randomness pseudorandomly from the address. In short, a VRF is a public key primitive which computes some random and unique output from an input together with a publicly verifiable (implicit) proof of correct computation. If one would have a VRF where the randomness itself is not leaked, but its output is a group element and only the holder of the VRF secret key knows the discrete logarithm of this group element with respect to the base element of the group, then this could work. Indeed, the Dodis-Yampolskiy (DY) construction [DY05] satisfies this property and additionally has compact keys and proofs.[9] While using such a VRF to derive the randomness for the signature scheme from the address seems intuitively secure, there does not seem to be a viable proof strategy to prove EUF-CMA security with a (black-box) reduction to the VRF and the signature scheme. The problem is that we see no way of decoupling the output of the VRF and the randomness in the signature scheme to come up with a working simulation strategy in the security proof. Even decoupling and proving consistency using NIZKs did not work for any strategy we tried. As we, moreover, do not want to resort on highly idealized models such as the generic group model [Sho97] to directly analyse such a DAPS construction (cf. Section 4.3 for problems with such an analysis for ECDSA), we pursue an alternative path where we can avoid such models use the signature scheme in a black-box fashion.

A working path. Besides the problems which turn up when pursuing the direction sketched above, it turns out to be highly non-trivial to achieve the desired functionality in the discrete logarithm setting in general. In particular, the requirement to be able to extract a certain discrete logarithm, i.e., the secret key, as soon as more than one signature within the same context exists, makes it very hard to perform the simulation within the security reduction when trying to relate the unforgeability of the DAPS to the unforgeability of the underlying signature scheme in a black-box fashion.

Fortunately, we are nevertheless able to come up with novel techniques which are inspired by secret sharing. In particular, we use a secret sharing of the secret signing key (in $\mathbb{Z}_q$) such that producing signatures for two colliding messages, i.e., messages with identical address but different payloads, allows to reconstruct the secret, i.e., the signing key. If now every address a is associated to a degree 1 polynomial $f_a(X)$ with $f_a(0)$ being the signing key and every signature includes a share $f_a(p)$ (evaluation of the polynomial on the payload p of the message to be signed), two colliding messages reveal the signing key. The tricky part

[9] We could even avoid bilinear groups in the DY VRF by providing an efficient NIZK of validity of the verification equation instead of using a pairing to check the proof.

is that one additionally requires a mechanisms to convince a verifier that the signer behaves honest, i.e., really reveals a share of the key associated to the address-polynomial, while still preserving the ability to conduct the simulation in the security reduction. While latter is typically approached by adding verifiability to the secret sharing scheme using a mapping of the coefficients defining $f_a(X)$ to the group $\mathcal{G} = (\mathbb{G}, q, g)$, we can not do so as this immediately destroys the possibility to conduct a black-box reduction to the $\mathsf{EUF\text{-}CMA}$ security of the underlying signature scheme (essentially the public verifiability destroys the possibility to simulate in the security proof).

To this end, we need a trick to decouple the public verifiability of the secret sharing from the signing key to make the proof work. We approach this by encrypting the coefficients of the address-polynomials mapped to elements of $\mathbb{G}$ (except the constant term representing the public key of the signature scheme) and provide a zero-knowledge proof of knowledge (using an efficient Σ-protocol made non-interactive via Fiat-Shamir) that the value $f_a(p)$ in the signature really represents an evaluation of the encrypted address-polynomial. While conducting such a proof would already be sufficient for a working scheme, we additionally observe that we can employ linearly homomorphic encryption (e.g., ElGamal) to do some pre-computations before we actually conduct the proof. This, in turn, makes our approach highly efficient.

In addition, we observe that our approach directly allows us to derive a generalization to N-times-authentication-preventing signatures (NAPS) for arbitrary $N > 2$ by using higher degree polynomials.

Efficiency of our approach. Our constructions yield short signatures and are practically efficient (which we extensively discuss in Section 7). For instance, constructing a DAPS from ECDSA implemented using the `prime256v1` elliptic curve yield a signature of size 160 byte, being roughly 2.5 times the size of conventional ECDSA signatures. Signing is roughly 3.8 times and verification 1.6 times of conventional ECDSA. On the platform we use for benchmarking, signing and verification require 0.23 and 0.35 ms respectively.

4 Signature Schemes

In this section we firstly present a formal model for the security of signature schemes. Secondly, we present the ECDSA signature scheme which we later use to instantiate our DAPS construction.

4.1 Formal Model

Definition 1 (Signature Scheme). *A signature scheme* Σ *is a triple* $(\mathsf{KGen}_\Sigma, \mathsf{Sign}_\Sigma, \mathsf{Verify}_\Sigma)$ *of PPT algorithms, which are defined as follows:*

$\mathsf{KGen}_\Sigma(1^\kappa)$: *This algorithm takes a security parameter* κ *as input and outputs a secret (signing) key* sk_Σ *and a public (verification) key* pk_Σ *with associated message space* $\mathcal{M}$ *(we may omit to make the message space* $\mathcal{M}$ *explicit).*

$\mathsf{Sign}_\Sigma(\mathsf{sk}_\Sigma, m)$: *This algorithm takes a secret key* sk_Σ *and a message* $m \in \mathcal{M}$ *as input and outputs a signature* σ*.*

$\mathsf{Verify}_\Sigma(\mathsf{pk}_\Sigma, m, \sigma)$: *This algorithm takes a public key* pk_Σ*, a message* $m \in \mathcal{M}$ *and a signature* σ *as input and outputs a bit* $b \in \{0, 1\}$*.*

We require a signature scheme to be correct and EUF-CMA secure. For correctness we require that for all $\kappa \in \mathbb{N}$, for all $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow \mathsf{KGen}_\Sigma(1^\kappa)$ and for all $m \in \mathcal{M}$ it holds that

$$\Pr\left[\mathsf{Verify}_\Sigma(\mathsf{pk}_\Sigma, m, \mathsf{Sign}_\Sigma(\mathsf{sk}_\Sigma, m)) = 1\right] = 1.$$

Definition 2 (EUF-CMA). *A signature scheme* Σ *is* EUF-CMA *secure, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\varepsilon(\cdot)$ *such that*

$$\Pr\left[\mathbf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\Sigma}(\kappa) = 1\right] \leq \varepsilon(\kappa),$$

where the corresponding experiment is depicted in Figure 1.

```
$\mathbf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\Sigma}(\kappa)$:
  $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow \mathsf{KGen}_\Sigma(1^\kappa)$
  $\mathcal{Q} \leftarrow \emptyset$
  $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}'_\Sigma(\mathsf{sk}_\Sigma, \cdot)}(\mathsf{pk})$
    where oracle $\mathsf{Sign}'_\Sigma$ on input $m$:
      let $\sigma \leftarrow \mathsf{Sign}_\Sigma(\mathsf{sk}_\Sigma, m)$
      set $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
      return $\sigma$
  return 1, if $\mathsf{Verify}_\Sigma(\mathsf{pk}_\Sigma, m^*, \sigma^*) = 1 \ \wedge \ m^* \notin \mathcal{Q}$
  return 0
```

Fig. 1: EUF-CMA **security.**

4.2 Elliptic Curve Groups

We briefly recall groups from elliptic curves. Let an elliptic curve E over the finite field $\mathbb{F}_p$ be a plane, smooth algebraic curve usually defined by a Weierstrass equation. The set $E(\mathbb{F}_p)$ of points $(x, y) \in \mathbb{F}_p^2$ satisfying this equation plus the point at infinity $\mathcal{O}$, which is the neutral element, forms an additive Abelian group, whereas the group law is determined by the chord-and-tangent method. If we write P_x we refer to the x coordinate of a point P. In general, we write $\mathcal{G} = (\mathbb{G}, q, g)$ to denote a group $\mathbb{G}$ of order q with generator g and we always use multiplicative notion throughout the paper.

```
$\underline{\mathsf{KGen}_{\mathsf{ECDSA}}(1^\kappa)}$: Let $\mathcal{G} = (\mathbb{G}, q, g)$ be an elliptic curve group. Choose $x \xleftarrow{R} \mathbb{Z}_q^*$ and set
    $\mathsf{sk} \leftarrow x$ and $\mathsf{pk} \leftarrow g^x$ and return $(\mathsf{sk}, \mathsf{pk})$.
$\underline{\mathsf{Sign}_{\mathsf{ECDSA}}(\mathsf{sk}, m)}$: Parse $\mathsf{sk}$ as $x$
        1. choose $k \xleftarrow{R} \mathbb{Z}_q^*$
        2. compute $R \leftarrow g^k$
        3. let $r \leftarrow R_x \pmod q$ and if $r = 0$ goto step 1
        4. let $s \leftarrow k^{-1}(H(m) + rx) \pmod q$ and if $s = 0$ goto step 1
        5. return $\sigma \leftarrow (r, s)$
$\underline{\mathsf{Verify}_{\mathsf{ECDSA}}(\mathsf{pk}, m, \sigma)}$: Parse $\sigma$ as $(r, s)$
        1. If $r = 0 \ \vee \ s = 0$ return 0
        2. let $z \leftarrow H(m)$ and $w \leftarrow s^{-1} \pmod q$
        3. let $u_1 \leftarrow zw \pmod q$ and $u_2 \leftarrow rw \pmod q$
        4. let $R \leftarrow g^{u_1} \cdot \mathsf{pk}^{u_2}$
        5. if $R_x = r \pmod q$ return 1 and return 0 otherwise
```

Scheme 1: ECDSA signature scheme.

4.3 ECDSA

In Scheme 1 we recall the ECDSA signature scheme. Thereby, $H : \{0,1\}^* \rightarrow \mathbb{Z}_q$ is a hash function mapping exactly to the order of the group.

The security analysis of ECDSA was for quite some time a topic of debates. There exist proofs of security of modified variants of ECDSA [MS02]. Brown [Bro02,Bro05] provides an analysis of standard ECDSA in the generic group model [Sho97], which quite leaves some open questions (cf. [FKP16] for a discussion why such a proof is problematic for ECDSA). The most recent work on the security of ECDSA from Fersch et al. [FKP16] avoids the generic group model and proves $\mathsf{EUF\text{-}CMA}$ security of ECDSA in the bijective random oracle model (ROM). We want to emphasize that we do not require details of any technique to prove security of ECDSA in this paper, as we will make a black-box reduction to $\mathsf{EUF\text{-}CMA}$ security of ECDSA.

5 Double-Authentication-Preventing Signatures

5.1 Formal Model

For double-authentication-preventing signatures (DAPS), we have a signature scheme on a message space $\mathcal{M} = \mathsf{A} \times \mathsf{P}$ of messages $m = (a, p)$ consisting of an address a and a payload p. The signature scheme is extended with a fourth algorithm Ex that extracts the secret key from signatures on two colliding messages. Before we can present the formal definition of DAPS we need to define the term colliding messages.

Definition 3 (Colliding Messages). *We call two messages $m_1 = (a_1, p_1)$ and $m_2 = (a_2, p_2)$ colliding if $a_1 = a_2$, but $p_1 \neq p_2$.*

Below, we now formally introduce DAPS following [PS14,PS17].

Definition 4 (DAPS). *A double-authentication-preventing signature scheme* DAPS *is a tuple* $(\mathsf{KGen}_\mathsf{D}, \mathsf{Sign}_\mathsf{D}, \mathsf{Verify}_\mathsf{D}, \mathsf{Ex}_\mathsf{D})$ *of PPT algorithms, which are defined as follows:*

$\mathsf{KGen}_\mathsf{D}(\kappa)$: *This algorithm takes a security parameter* κ *as input and outputs a secret (signing) key* sk_D *and a public (verification) key* pk_D *with associated message space* $\mathcal{M}$ *(we may omit to make the message space* $\mathcal{M}$ *explicit).*

$\mathsf{Sign}_\mathsf{D}(\mathsf{sk}_\mathsf{D}, m)$: *This algorithm takes a secret key* sk_D *and a message* $m \in \mathcal{M}$ *as input and outputs a signature* σ*.*

$\mathsf{Verify}_\mathsf{D}(\mathsf{pk}_\mathsf{D}, m, \sigma)$: *This algorithm takes a public key* pk_D*, a message* $m \in \mathcal{M}$ *and a signature* σ *as input and outputs a bit* $b \in \{0, 1\}$*.*

$\mathsf{Ex}_\mathsf{D}(\mathsf{pk}_\mathsf{D}, m_1, m_2, \sigma_1, \sigma_2)$: *This algorithm takes a public key* pk_D*, two colliding messages* m_1 *and* m_2 *and signatures* σ_1 *for* m_1 *and* σ_2 *for* m_2 *as inputs and outputs a secret key* sk_D*.*

Note that the algorithms KGen_D, Sign_D, and $\mathsf{Verify}_\mathsf{D}$ match the definition of the algorithms of a conventional signature scheme. For DAPS one requires a restricted but otherwise standard notion of unforgeability [PS14,PS17], where adversaries can adaptively query signatures for messages but only on distinct addresses. Figure 2 details the unforgeability security experiment.

Definition 5 (EUF-CMA [**PS14**]). *A DAPS scheme is* EUF-CMA *secure, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\varepsilon(\cdot)$ *such that*

$$\Pr\left[\mathbf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = 1\right] \leq \varepsilon(\kappa),$$

where the corresponding experiment is depicted in Figure 2.

```
$\mathbf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\mathsf{DAPS}}(\kappa)$:
  $(\mathsf{sk}_\mathsf{D}, \mathsf{pk}_\mathsf{D}) \leftarrow \mathsf{KGen}_\mathsf{D}(1^\kappa)$
  $\mathcal{Q} \leftarrow \emptyset$, $\mathcal{R} \leftarrow \emptyset$
  $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}'_\mathsf{D}(\mathsf{sk}_\mathsf{D}, \cdot)}(\mathsf{pk}_\Sigma)$
    where oracle $\mathsf{Sign}'_\mathsf{D}$ on input $m$:
      $(a, p) \leftarrow m$
      if $a \in \mathcal{R}$, return $\perp$
      $\sigma \leftarrow \mathsf{Sign}_\mathsf{D}(\mathsf{sk}_\mathsf{D}, m)$
      $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$, $\mathcal{R} \leftarrow \mathcal{R} \cup \{a\}$
      return $\sigma$
  return 1, if $\mathsf{Verify}_\mathsf{D}(\mathsf{pk}_\mathsf{D}, m^*, \sigma^*) = 1 \ \wedge \ m^* \notin \mathcal{Q}$
  return 0
```

Fig. 2: EUF-CMA **security for** DAPS.

The interesting property of a DAPS scheme is the notion of double-signature extractability (DSE). It requires that whenever one obtains signatures on two colliding messages, one should be able to extract the signing key using the extraction algorithm Ex_D. We give the security game in Figure 3, where we consider the conventional notion, denoted as DSE, which requires extraction to work if the key pair has been generated honestly. In this game, the adversary is given a key pair and outputs two colliding messages and corresponding signatures. The adversary wins the game if the key produced by Ex_D is different from the signing key although extraction should have succeeded, i.e, the messages were colliding and their signatures were valid.

Definition 6 (DSE [PS14]). *A DAPS scheme provides double-signature extraction (*DSE*), if for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that*

$$\Pr\left[\mathbf{Exp}^{\mathsf{DSE}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = 1\right] \leq \varepsilon(\kappa),$$

where the corresponding experiment is depicted in Figure 3.

```
$\mathbf{Exp}^{\mathsf{DSE}}_{\mathcal{A},\mathsf{DAPS}}(\kappa)$:
  $(\mathsf{sk}_\mathsf{D}, \mathsf{pk}_\mathsf{D}) \leftarrow \mathsf{KGen}_\mathsf{D}(1^\kappa)$
  $(m_1, m_2, \sigma_1, \sigma_2) \leftarrow \mathcal{A}(\mathsf{sk}_\mathsf{D}, \mathsf{pk}_\mathsf{D})$
  return 0, if $m_1$ and $m_2$ are not colliding
  $v_i \leftarrow \mathsf{Verify}_\mathsf{D}(\mathsf{pk}_\mathsf{D}, m_i, \sigma_i)$ for $i \in [2]$
  return 0, if $v_1 = 0$ or $v_2 = 0$
  $\mathsf{sk}'_\mathsf{D} \leftarrow \mathsf{Ex}_\mathsf{D}(\mathsf{pk}_\mathsf{D}, m_1, m_2, \sigma_1, \sigma_2)$
  return 1, if $\mathsf{sk}'_\mathsf{D} \neq \mathsf{sk}_\mathsf{D}$
  return 0
```

Fig. 3: DSE **security for** DAPS**.**

In Appendix C we recall the strong variant of extractability under malicious keys (denoted as DSE*), where the adversary is allowed to generate the key arbitrarily. The DSE* notion is very interesting from a theoretical perspective, but no efficient DAPS construction, including ours, can achieve this notion so far. However, as we will show in Section 6.5 our, constructions satisfy a weaker notion under malicious keys introduced in this paper.

5.2 Existing DAPS Constructions

Poettering and Stebila [PS14,PS17] present the first ever DAPS construction in a factoring-based setting, where a signature contains $n+1$ elements in a group $\mathbb{Z}_N^*$ with n being the length of the output of a cryptographic hash function and N is an RSA modulus. At a security level of 128 bit (a 2048-bit RSA modulus and 256-bit hash), a signature contains > 250 group elements yielding a signature size of > 64 KB and signing as well as verification times much higher than standard signatures. Ruffing, Kate and Schroeder in [RKS15] introduced the notion of accountable assertions (AS), a weaker primitive than DAPS, and present one AS that also is a DAPS (termed RKS). The RKS construction is based on Merkle tress and chameleon hash functions in the discrete logarithm setting. Signing and verification are much more efficient than within PS, but signature sizes are still in the order of PS. Very recently, Bellare, Poettering and Stebila [BPS17] proposed new factoring-based DAPS from trapdoor identification-schemes using an adaption and extension of a transform from [BPS16]. Their two transforms applied to the Guillou-Quisquater (GQ) [GQ88] and Micali-Reyzin (MR) [MR02] identification scheme yield signing and verification times as well as signature sizes comparable (or slightly above) standard RSA signatures. In a concurrent and independent work Boneh et al. [BKN17] propose constructions of DAPS from lattices. They consider DAPS as a special case of what they call predicate-authentication-preventing signatures (PAPS). In PAPS one considers a k-ary predicate on the message space and given any k valid signatures that satisfy the predicate reveal the signing key. Consequently, DAPS are PAPS for a specific 2-ary predicate and what we call N-times-authentication-preventing signatures (NAPS) is denoted as k-way DAPS in their work.

Unfortunately, as it is clear from the discussion, none of these DAPS schemes relies on widely used signature schemes such as RSA or (EC)DSA signatures. It is also important to mention that all these constructions only provide the extractability notion under honestly generated keys (DSE)[10]. We now present our DAPS in the next section and defer a detailed comparison of existing DAPS and ours to Section 6.10.

6 Short DAPS in the DL Setting

In this section we present our generic DAPS constructions from any discrete logarithm-based $\mathsf{EUF\text{-}CMA}$ secure signature scheme and in particular provide an instantiation with ECDSA signatures. As already mentioned, we thereby will be as non-invasive as possible in constructing DAPS "around" existing signatures without modifying the setting, e.g., groups, that are used by the respective schemes.

6.1 Intuition of Our Approach

As already mentioned in Section 3, our generic approach to construct DAPS is based on the idea of combining a signature scheme with a verifiable secret sharing scheme and in every signature include a share (specific to the address) of the secret signing key. Consequently, signing two different payloads with respect to the same address within the DAPS allows to extract the signing key of the underlying signature scheme.

[10] To be precise, in the initial work [PS14,PS17] the authors could tweak their construction to provide DSE* at the cost of adding quite expensive non-interactive zero-knowledge proofs to show that the public key is a well-formed Blum integer. But this would make their already rather impractical constructions with signature sizes $>$ 64 KB only more impractical.

Before presenting our construction paradigm and instantiations of DAPS, we introduce verifiable secret sharing in Section 6.2, ElGamal encryption in Section 6.3 and non-interactive zero-knowledge proofs from Σ-protocols (and a standard proof for the language of DDH tuples) in Section 6.4.

6.2 Verifiable Secret Sharing

Shamir's (k, ℓ)-threshold secret sharing [Sha79] allows to information-theoretically share a secret s among ℓ parties such that whenever k evaluations of the polynomial (shares) are given, reconstruction of s is possible, but as long as only $k-1$ shares are available the secret s is information-theoretically hidden. Let s be the constant term of an otherwise randomly chosen $k-1$ degree polynomial

$$f(X) = \rho_{k-1}X^{k-1} + \cdots + \rho_1 X + s$$

over a prime field $\mathbb{Z}_q$. A share is computed as $f(i)$ for party i, $1 \leq i \leq \ell$. Let $\mathcal{S}$ be any set of cardinality at least k of these ℓ shares and let us denote the set of indices corresponding to shares in $\mathcal{S}$ by $I_{\mathcal{S}}$. Using Lagrange interpolation one can compute $s = f(0)$ as

$$s = \sum_{j \in I_{\mathcal{S}}} \lambda_j f(j) \text{ whereas } \lambda_j = \prod_{i \in I_{\mathcal{S}} \setminus \{j\}} \frac{j}{j-i} \ .$$

Now, we discuss a well known technique due to Feldman [Fel87] to make Shamir's secret sharing verifiable, by relaxing the otherwise information-theoretic secrecy to be only computational. The basic idea is to allow the use of a one-way homomorphism and in particular let us use a group $\mathcal{G} = (\mathbb{G}, q, g)$. To enable verifiability one publishes the sequence $(g^{\rho_{k-1}}, \ldots, g^{\rho_1}, g^{\rho_0})$ with $g^{\rho_0} = g^s$ and when given a share $f(i)$, everyone can non-interactively verify whether the share is correct by checking

$$g^{f(i)} = \prod_{j=0}^{k-1} (g^{\rho_j})^{i^j} .$$

Clearly, secrecy of s is only guaranteed if it has high min-entropy, as guesses can efficiently be verified.

6.3 ElGamal Encryption

Before presenting ElGamal encryption [Gam84], let us define an encryption scheme first.

Definition 7 (Public Key Encryption Scheme). *A public key encryption scheme Ω is a triple* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *of PPT algorithms such that:*

$\mathsf{KGen}(1^\kappa)$: *This algorithm on input security parameter κ outputs the secret and public key* $(\mathsf{sk}, \mathsf{pk})$ *(the public key* pk *implicitly defines the message space $\mathcal{M}$).*

$\mathsf{Enc}(\mathsf{pk}, m)$: *This algorithm input the public key* pk*, and the message* $m \in \mathcal{M}$ *and outputs a ciphertext* C.

$\mathsf{Dec}(\mathsf{sk}, C)$: *This algorithm on input a secret key* sk *and a ciphertext* C *outputs a message* $m \in \mathcal{M} \cup \{\perp\}$.

We say that an encryption scheme Ω is perfectly correct if for all $\kappa \in \mathbb{N}$, for all $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\kappa)$ and for all $m \in \mathcal{M}$ it holds that

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m\right] = 1.$$

IND-CPA security requires that an adversary $\mathcal{A}$ cannot decide which message is actually contained in a ciphertext C even when allowed to choose two challenge messages m_0 and m_1. We formally define IND-CPA security in Appendix D.

The ElGamal encryption scheme is multiplicatively homomorphic and IND-CPA secure under the k-LIN assumption in $\mathcal{G}$. We briefly present the popular ElGamal encryption scheme [Gam84] in a group $\mathcal{G} = (\mathbb{G}, q, g)$ where the 1-LIN (DDH) assumption holds. The key generation algorithm KGen on input κ generates a group $\mathcal{G} = (\mathbb{G}, q, g)$ of prime order q of size κ bits and sets $sk := x \overset{R}{\leftarrow} \mathbb{Z}_q$ and $\mathsf{pk} := g^x$. To encrypt a message $m \in \mathbb{G}$, Enc samples $r \overset{R}{\leftarrow} \mathbb{Z}_q$ and computes the ciphertext $(C_1, C_2) := (g^r, m \cdot \mathsf{pk}^r)$. Finally, the decryption algorithm Dec given sk and ciphertext (C_1, C_2) outputs $C_2 \cdot C_1^{-\mathsf{sk}}$.

When setting $k = 2$ instead of $k = 1$ one obtains ElGamal under the 2-LIN (DLIN) assumption [BBS04] (termed linear ElGamal). It has the benefit that it can be instantiated in groups where the DDH assumption does not hold, e.g., in certain pairing-friendly elliptic curve or Schnorr groups. We recall both assumptions in Appendix A for the convenience of the reader. In the remainder of this paper we use the DDH instantiation of ElGamal, but we stress that all our protocols can be based on linear ElGamal as well.

6.4 Σ-Protocols

Let $L \subseteq \mathsf{X}$ be an **NP**-language with associated witness relation R so that $L = \{x \mid \exists w : R(x, w) = 1\}$. A Σ-protocol for language L is an interactive three move protocol between a prover and a verifier, where the prover proves knowledge of a witness w to the statement $x \in L$. We recall the formal definition of Σ-protocols in Appendix E.

Σ-protocol for DDH-tuples. Σ-protocols for proving that elements (g_1, g_2, u_1, u_2) in a prime order group $\mathcal{G}$ form a DDH tuple are well known and established [CP92]. We define the corresponding language via relation R

$$((g_1, g_2, u_1, u_2), w) \in R \Leftrightarrow g_1^w = u_1 \wedge g_2^w = u_2 \tag{1}$$

as witness relation. In Scheme 2 we briefly recall a classical Σ-protocol for R.

Lemma 1. *The protocol in Scheme 2 represents a Σ-protocol for the relation R in* (1).

Let $\mathcal{G} = (\mathbb{G}, q, g)$ and let $g_1, g_2, u_1, u_2 \in \mathbb{G}$.

Prover		**Verifier**
$(u_1, u_2, k = \log_{g_i} u_i)$		(u_1, u_2)
$r \xleftarrow{R} \mathbb{Z}_q^*,\ r_i \leftarrow g_i^r$	$\xrightarrow{r_1, r_2}$	
	$\xleftarrow{c}$	$c \xleftarrow{R} \mathbb{Z}_q$
$s \leftarrow r + kc$	$\xrightarrow{s}$	accept iff $\forall i : g_i^s = r_i u_i^c$

Scheme 2: Σ-protocol for proving that (g_1, g_2, u_1, u_2) forms a DDH-tuple.

We omit the proof of Lemma 1 as it is a well known result and straightforward.

Non-Interactive ZK Proof Systems (NIZK). We recall a standard definition of non-interactive zero-knowledge proof systems. Let L be an **NP**-language with witness relation R as above.

Definition 8 (Non-Interactive Zero-Knowledge Proof System). *A non-interactive proof system* Π *is a tuple of algorithms* $(\mathsf{Setup}_\Pi, \mathsf{Proof}_\Pi, \mathsf{Verify}_\Pi)$*, which are defined as follows:*

$\mathsf{Setup}_\Pi(1^\kappa)$: *This algorithm takes a security parameter* κ *as input, and outputs a common reference string* crs.

$\mathsf{Proof}_\Pi(\mathsf{crs}, x, w)$: *This algorithm takes a common reference string* crs*, a statement* x*, and a witness* w *as input, and outputs a proof* π.

$\mathsf{Verify}_\Pi(\mathsf{crs}, x, \pi)$: *This algorithm takes a common reference string* crs*, a statement* x*, and a proof* π *as input, and outputs a bit* $b \in \{0, 1\}$.

From a non-interactive zero-knowledge proof system we require *completeness*, *soundness* and *adaptive zero-knowledge*. In Appendix F we recall formal definitions of those properties.

NIZK from Σ-protocols. One can obtain a non-interactive proof system with the above properties from any Σ-protocol by applying the Fiat-Shamir transform [FS86] where the min-entropy μ of the commitment a sent in the first message of the Σ-protocol is so that $2^{-\mu}$ is negligible in the security parameter κ and its challenge space C is exponentially large in the security parameter. Essentially, the transform removes the interaction between the prover and the verifier by using a hash function H (modelled as a random oracle) to obtain the challenge. That is, the algorithm $\mathsf{Challenge}$ obtains the challenge as $H(\mathsf{a}, x)$. More formally, $\mathsf{Setup}_\Pi(1^\kappa)$ fixes a hash function $H : \mathsf{A} \times \mathsf{X} \to \mathsf{C}$, sets $\mathsf{crs} \leftarrow (\kappa, H)$ and returns crs. The algorithms Proof_Π and Verify_Π are defined as follows:

$\mathsf{Proof}_\Pi(\mathsf{crs}, x, w)$: Start P on $(1^\kappa, x, w)$, obtain the first message a, answer with $\mathsf{c} \leftarrow H(\mathsf{a}, x)$. Finally obtain s and return $\pi \leftarrow (\mathsf{a}, \mathsf{s})$.

$\mathsf{Verify}_{\Pi}(\mathsf{crs}, x, \pi)$: Parse π as (a, s). Start V on $(1^{\kappa}, x)$ and send a as first message to the verifier. When V outputs c, reply with s and output 1 if V accepts and 0 otherwise.

Combining [FKMV12, Thm. 1, Thm. 2, Thm. 3, Prop. 1] (among others) shows that a so-obtained proof system is complete, sound, adaptively zero-knowledge, if the underlying Σ-protocol is special sound and the commitments sent in the first move are unconditionally binding. When referring to the NIZK proof system obtained from Scheme 2, we denote the algorithms as $(\mathsf{Setup}_{\mathsf{DDH}}, \mathsf{Proof}_{\mathsf{DDH}}, \mathsf{Verify}_{\mathsf{DDH}})$.

A note on the CRS. We stress that for the sake of generality the output of $\mathsf{Setup}_{\mathsf{DDH}}$ is denoted as crs. However, as we exclusively use NIZK from Σ-protocols in our DAPS, we do not require a trusted setup and crs is just a description of the hash function which is globally fixed, e.g., to SHA-256 or SHA-3.

6.5 Extraction of the Signing Key of $\mathbf{\Sigma}$

When considering constructions that extend conventional signature schemes to a DAPS, there is a gap between DSE and DSE^* notions and ensuring extraction of the $\mathsf{\Sigma}$ signing key. Recall, that these notions require to extract the complete DAPS secret key and no existing efficient DAPS scheme provides DSE^*. When the DAPS key consists of a $\mathsf{\Sigma}$ signing key, extraction of the signing key alone, however, already disincentivizes double-authentication for many applications, where this key is also used outside the context of DAPS. Hence we define two weaker double-signature extraction notions that cover extraction of the signing key of the underlying signature scheme for honestly and maliciously generated DAPS keys. The security games for weak double-signature extraction (wDSE) and weak double-signature extraction under malicious keys (wDSE^*) are depicted in Figure 4 and Figure 5.

Definition 9 **($T \in \{\mathsf{wDSE}, \mathsf{wDSE}^*\}$).** *A DAPS scheme provides weak double-signature extraction ($T = \mathsf{wDSE}$) respectively weak double-signature extraction under malicious keys ($T = \mathsf{wDSE}^*$), if for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that*

$$\Pr\left[\mathbf{Exp}^{T}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = 1\right] \leq \varepsilon(\kappa),$$

where the corresponding experiments are depicted in Figure 4 and Figure 5 respectively.

Clearly, DSE and DSE^* imply their weaker counterparts and wDSE^* implies wDSE.

```
$\mathbf{Exp}^{\mathsf{wDSE}}_{\mathcal{A},\mathsf{DAPS}}(\kappa)$:
  $(\mathsf{sk_D}, \mathsf{pk_D}) \leftarrow \mathsf{KGen_D}(1^\kappa)$ with $\mathsf{sk_D} = (\mathsf{sk}_\Sigma, \dots)$
  $(m_1, m_2, \sigma_1, \sigma_2) \leftarrow \mathcal{A}(\mathsf{sk_D}, \mathsf{pk_D})$
  return 0, if $m_1$ and $m_2$ are not colliding
  $v_i \leftarrow \mathsf{Verify_D}(\mathsf{pk_D}, m_i, \sigma_i)$ for $i \in [2]$
  return 0, if $v_1 = 0$ or $v_2 = 0$
  $\mathsf{sk}'_\mathsf{D} \leftarrow \mathsf{Ex_D}(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2)$ where $\mathsf{sk}'_\mathsf{D} = (\mathsf{sk}'_\Sigma, \dots)$
  return 1, if $\mathsf{sk}'_\Sigma \neq \mathsf{sk}_\Sigma$
  return 0
```

Fig. 4: wDSE **security for** DAPS**.**

```
$\mathbf{Exp}^{\mathsf{wDSE}^*}_{\mathcal{A},\mathsf{DAPS}}(\kappa)$:
  $(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2) \leftarrow \mathcal{A}(1^\kappa)$ where $\mathsf{pk_D} = (\mathsf{pk}_\Sigma, \dots)$
  return 0, if $m_1$ and $m_2$ are not colliding
  $v_i \leftarrow \mathsf{Verify_D}(\mathsf{pk_D}, m_i, \sigma_i)$ for $i \in [2]$
  return 0, if $v_1 = 0$ or $v_2 = 0$
  $\mathsf{sk}'_\mathsf{D} \leftarrow \mathsf{Ex_D}(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2)$ where $\mathsf{sk}'_\mathsf{D} = (\mathsf{sk}'_\Sigma, \dots)$
  return 1, if $\mathsf{sk}'_\Sigma$ is not the secret key corresponding to $\mathsf{pk}_\Sigma$
  return 0
```

Fig. 5: wDSE* **security for** DAPS**.**

6.6 Generic DAPS in the Discrete Logarithm Setting

In the following, let Σ be a signature scheme in the discrete logarithm setting, which is from the class C of signature schemes where the public key is the image of the secret key under a group homomorphism. In the discrete logarithm setting this means that the secret key x is an element from $\mathbb{Z}_q$ and the public key is its image g^x in the group. We stress that the class C essentially covers any scheme in the discrete logarithm setting we can think of, and, in particular schemes like Schnorr, (EC)DSA, or EdDSA. We subsequently present our protocols based on ElGamal in the DDH setting and recall that when the DDH is not hard in the respective group, we can easily instantiate all our protocols on linear ElGamal under the DLIN assumption (cf. Section 6.3)

Our approach is as follows. First we generate an ElGamal encryption keypair (x_E, pk_E). Then, for each possible address i we choose $\rho_i \in \mathbb{Z}_q$ uniformly at random and additionally include an encryption $(C_{i,1}, C_{i,2})$ of g^{ρ_i} as well as pk_E in the DAPS public key. The secret key additionally includes the values ρ_i and the randomness $r_i \in \mathbb{Z}_q$ used upon encrypting ρ_i. When signing a message $m = (i, p) \in [n] \times \mathbb{Z}_q^*$, we obtain a signature from Σ, and extend it with a secret share of sk_Σ: we let $f_i(X) = \rho_i X + \mathsf{sk}_\Sigma$ and include $z = f_i(p)$ in the signature. When signing two colliding messages, we obtain two shares for the same degree 1 polynomial f_i and hence can re-construct sk_Σ. To ensure the correct computation of z, each signature is extended by a proof for the following relation R, which is essentially a proof for a verifiable secret sharing using ElGamal encryption for the coefficient of the non-constant term:

$$((g, \mathsf{pk}_E, C_{i,1}, C'_{i,2}), r) \in R \Leftrightarrow C_{i,1} = g^r \wedge C'_{i,2} = \mathsf{pk}_E^r$$

where $C'_{i,2} = C_{i,2} \cdot (\mathsf{pk}_\Sigma \cdot g^{-z})^{1/p}$.

Observe that the extraction algorithm, when applied to colliding signatures, reveals the secret signing key sk_Σ, but none of the r_i and ρ_i. However, DAPS extraction needs to recover the full secret key, so we cannot achieve the stronger DSE notion, but obtain wDSE security.

```
$\underline{\mathsf{KGen}_\mathsf{D}(1^\kappa, n)}$: Let $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow \mathsf{KGen}_\Sigma(1^\kappa)$ with $\mathcal{G} = (\mathbb{G}, q, g)$. Let $x_E \xleftarrow{R} \mathbb{Z}_q^*$ and
    $\mathsf{pk}_E \leftarrow g^{x_E}$. Let $(\rho_i)_{i\in[n]} \xleftarrow{R} (\mathbb{Z}_q^*)^n$ and $(r_i)_{i\in[n]} \xleftarrow{R} (\mathbb{Z}_q^*)^n$. Set $(C_i)_{i\in[n]} \leftarrow$
    $(g^{r_i}, \mathsf{pk}_E^{r_i} g^{\rho_i})_{i\in[n]}$. Let $\mathsf{crs} \leftarrow \mathsf{Setup}_\mathsf{DDH}(1^\kappa)$. Let $\mathsf{sk} \leftarrow (\mathsf{sk}_\Sigma, (r_i, \rho_i)_{i\in[n]})$ and
    $\mathsf{pk} \leftarrow (\mathsf{pk}_\Sigma, \mathsf{pk}_E, (C_i)_{i\in[n]}, \mathsf{crs})$ and return $(\mathsf{sk}, \mathsf{pk})$.
$\underline{\mathsf{Sign}_\mathsf{D}(\mathsf{sk}, m)}$: Parse $\mathsf{sk}$ as $(\mathsf{sk}_\Sigma, (r_i, \rho_i)_{i\in[n]})$. Parse $m$ as $(i, p)$ with $i \leq n$ and $p \in \mathbb{Z}_q^*$.
        1. Let $\sigma \leftarrow \mathsf{Sign}_\Sigma(\mathsf{sk}_\Sigma, m)$
        2. let $z \leftarrow \rho_i p + \mathsf{sk}_\Sigma$
        3. let $C'_2 \leftarrow C_{i,2} \cdot (\mathsf{pk}_\Sigma \cdot g^{-z})^{\frac{1}{p}}$
        4. $\pi \leftarrow \mathsf{Proof}_\mathsf{DDH}(\mathsf{crs}, (g, \mathsf{pk}_E, C_{i,1}, C'_2), r_i)$
        5. return $(\sigma, z, \pi)$
$\underline{\mathsf{Verify}_\mathsf{D}(\mathsf{pk}, m, \sigma)}$: Parse $\mathsf{pk}$ as $(\mathsf{pk}_\Sigma, \mathsf{pk}_E, (C_i)_{i\in[n]}, \mathsf{crs})$, $m$ as $(i, p)$ with $i \leq n$, and $\sigma$ as
    $(\sigma', z, \pi)$.
        1. If $\mathsf{Verify}_\Sigma(\mathsf{pk}_\Sigma, m, \sigma') = 0$, return 0
        2. let $C'_2 \leftarrow C_{i,2} \cdot (\mathsf{pk}_\Sigma \cdot g^{-z})^{\frac{1}{p}}$
        3. return $\mathsf{Verify}_\mathsf{DDH}(\mathsf{crs}, (g, \mathsf{pk}_E, C_{i,1}, C'_2), \pi)$
$\underline{\mathsf{Ex}_\mathsf{D}(\mathsf{pk}, m_1, m_2, \sigma_1, \sigma_2)}$: Parse $\sigma_i$ as $(\cdot, z_i, \cdot)$, $m_i$ as $(a_i, p_i)$ and $\mathsf{pk}$ as $(\cdot, \cdot, \cdot, \cdot)$.
        1. If $m_1$ and $m_2$ are not colliding, return $\perp$
        2. if $\mathsf{Verify}_\mathsf{D}(\mathsf{pk}, m_i, \sigma_i) = 0$ for any $i$, return $\perp$
        3. let $\mathsf{sk}_\Sigma \leftarrow z_1 \frac{p_2}{p_2 - p_1} + z_2 \frac{p_1}{p_1 - p_2}$
        4. return $\mathsf{sk}_\Sigma$
```

Scheme 3: Σ-DAPS: Generic DAPS from any signature scheme Σ from class C.

We note that in our construction KGen_D takes the number of addresses as explicit argument. The scheme is also presented using $\mathbb{Z}_q^*$ as payload space, but it can be extended to an arbitrary payload space using the standard hash-then-sign technique.

Theorem 1. *If Σ is from class* C *instantiated in group $\mathcal{G}$ and* EUF-CMA*-secure,* DDH *is hard relative to $\mathcal{G}$ and the NIZK proof system is adaptive zero-knowledge, then* Σ-DAPS *is an* EUF-CMA*-secure DAPS.*

Proof. We prove this theorem using a sequence of games. We denote the winning event of game G_i as S_i. We use gray textboxes to indicate changes within algorithms.

Game 0: The original $\mathsf{EUF\text{-}CMA}$ game.

Game 1: As before, but we modify KGen_D to use setup algorithm $\mathcal{S}_{1,\mathsf{DDH}}$ of the simulator for the NIZK proof system.

$\underline{\mathsf{KGen}_\mathsf{D}(1^\kappa, n)}$: As before, but let

$$\boxed{(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_{1,\mathsf{DDH}}(1^\kappa)} \text{ and store } \boxed{\tau}.$$

Transition $0 \to 1$**:** Game 0 and Game 1 are indistinguishable under adaptive zero-knowledge of the proof system, i.e. $|\Pr[S_0] - \Pr[S_1]| \leq \varepsilon_{z,1}(\kappa)$.

Game 2: As Game 1, but we modify Sign_D to use the simulation algorithm $\mathcal{S}_{2,\mathsf{DDH}}$ of the simulator of the NIZK proof system:

$\underline{\mathsf{Sign}_\mathsf{D}(\mathsf{sk}, m)}$: As before, but let

$$\boxed{\pi \leftarrow \mathcal{S}_{2,\mathsf{DDH}}(\mathsf{crs}, \tau, (g, \mathsf{pk}_E, C_{i,1}, C_2'))}.$$

Transition $1 \to 2$**:** Game 1 and Game 2 are indistinguishable under adaptive zero-knowledge of the proof system, i.e. $|\Pr[S_0] - \Pr[S_1]| \leq \varepsilon_{z,2}(\kappa)$.

Game 3: As Game 2, but we modify KGen_D as follows:

$\underline{\mathsf{KGen}_\mathsf{D}(1^\kappa, n)}$: Let $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow \mathsf{KGen}_\Sigma(1^\kappa)$ with $\mathcal{G} = (\mathbb{G}, q, g)$. Let $\boxed{\mathsf{pk}_E \xleftarrow{R} \mathbb{G}}$. Let $(\rho_i)_{i\in[n]} \xleftarrow{R} (\mathbb{Z}_q^*)^n$. Let $\boxed{(C_i)_{i\in[n]} \xleftarrow{R} (\mathbb{G}^2)^n}$. Let $(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_{1,\mathsf{DDH}}(1^\kappa)$. Let $\mathsf{sk} \leftarrow (\mathsf{sk}_\Sigma, (r_i, \rho_i)_{i\in[n]})$ and $\mathsf{pk} \leftarrow (\mathsf{pk}_\Sigma, \mathsf{pk}_E, (C_i)_{i\in[n]}, \mathsf{crs})$ and return $(\mathsf{sk}, \mathsf{pk})$.

Transition $2 \to 3$**:** We claim that the probability to distinguish between Game 1 and Game 2 is bounded by $|\Pr[S_1] - \Pr[S_2]| \leq n \cdot \varepsilon_\mathsf{DDH}(\kappa)$. To see this assume n additional hybrids, where in each hybrid H_j with $1 \leq j \leq n$ we replace ciphertext C_j by a random value. Then the distinguishing probability of two consecutive hybrids is bounded by $\varepsilon_\mathsf{DDH}(\kappa)$. In particular, assume we obtain a DDH instance $(g^{u_1}, g^{u_2}, g^{u_3})$ relative to $\mathbb{G}$ and set $\mathsf{pk}_E \leftarrow g^{u_2}$. Then in hybrid H_j we choose all C_i where $i < j$ random (as they were also already random in the previous hybrid). For C_j, we compute $C_j \leftarrow (g^{u_1}, g^{u_3} \cdot g^{\rho_i})$. Furthermore, for C_i where $i > j$, we choose $r_i \xleftarrow{R} \mathbb{Z}_q$ and set $C_i \leftarrow (g^{r_i}, (g^{u_2})^{r_i} \cdot g^{\rho_i})$. Then the validity of the DDH instance determines whether we sample from the distribution in Game i or Game $i+1$, which proves that the distinguishing probability between two intermediate hybrids is bounded by $\varepsilon_\mathsf{DDH}(\kappa)$. Taking all n transitions together, this yields $n \cdot \varepsilon_\mathsf{DDH}(\kappa)$ which proves our initial claim.

Game 4: As Game 3, but we modify Sign_D as follows:

$\underline{\mathsf{Sign}_\mathsf{D}(\mathsf{sk}, m)}$: As before, but let $\boxed{z \xleftarrow{R} \mathbb{Z}_q}$.

Transition $3 \to 4$**:** This change is conceptual. At this point sk_Σ is information-theoretically hidden.

Game 5: As Game 4, but we abort whenever the adversary comes up with a valid forgery.

Transition $4 \to 5$**:** We denote the event that we abort by E. Both, Game 4 and Game 5 proceed identically unless E happens, i.e., $|\Pr[S_2] - \Pr[S_3]| \leq \Pr[E]$. Whenever E happens in Game 5, we can build an $\mathsf{EUF\text{-}CMA}$ forger for Σ.

To do so, we engage with an $\mathsf{EUF\text{-}CMA}$ challenger for Σ and obtain σ from the oracle provided by the challenger (we no longer require sk_Σ anywhere else). If the adversary outputs a forgery, we can output $(\sigma', (i, m))$ as a valid $\mathsf{EUF\text{-}CMA}$ forgery, i.e. $|\Pr[S_2] - Pr[S_3]| \leq \varepsilon_{\mathsf{EUF\text{-}CMA}(\kappa)}$.

In the final game, the adversary can no longer win, i.e., $\Pr[S_5] = 0$. Taking all together, we have that $\Pr[S_0] \leq \varepsilon_{z,1}(\kappa) + \varepsilon_{z,2}(\kappa) + n \cdot \varepsilon_{\mathsf{DDH}}(\kappa) + \varepsilon_{\mathsf{EUF\text{-}CMA}}(\kappa)$, which concludes the proof.

We now show that our Σ-DAPS also provide wDSE security, and then extend this result to wDSE*, and thus for the first time we have some reasonable extraction guarantees under adversarially generated keys for practical DAPS.

Theorem 2. *If the NIZK proof system is sound, then* Σ-DAPS *provides* wDSE *security.*

Proof. We prove this theorem using a sequence of games. We denote the winning event of game G_i as S_i. Let $m_1, m_2, \sigma_1, \sigma_2$ be the output of $\mathcal{A}$. For simplicity we write $m_j = (a, p_j)$, $\sigma_j = (\cdot, z_j, \pi_j)$ for $i \in [2]$, $\mathsf{pk}_\mathsf{D} = (\mathsf{pk}_\Sigma, \mathsf{pk}_E, (C_i)_{i \in [n]}, \mathsf{crs})$, and $(C_{a,1}, C_{a_2}) \leftarrow C_a$. We also let $C'_{j,2} \leftarrow C_{a,2} \cdot (\mathsf{pk}_\Sigma \cdot g^{-z_j})^{\frac{1}{p_j}}$ for $j \in [2]$.

Game 0: The original wDSE game.
Game 1: As before, but we abort if $C'_{1,2} \neq C'_{2,2}$.
Transition $0 \to 1$**:** Let E be the event that $C'_{1,2} \neq C'_{2,2}$. In this case we engage with a soundness challenger $\mathcal{C}$ of proof system and modify KGen_D as follows:

$\underline{\mathsf{KGen}_\mathsf{D}(1^\kappa, n)}$: Obtain $\boxed{\mathsf{crs}}$ from $\mathcal{C}$ and compute everything else honestly.

Once $\mathcal{A}$ outputs the two colliding messages and signatures, we have proofs attesting that both $(g, \mathsf{pk}_E, C_{a,1}, C'_{j,2})$ for $j \in [2]$ are DDH tuples, but, by the perfect correctness of ElGamal, at most one of them can be a DDH tuple, i.e., one of the words is not in the language. Hence we guess $b \xleftarrow{R} \{0, 1\}$, and forward $(g, \mathsf{pk}_E, C_{a,1}, C'_{b+1,2}), \pi_{b+1}$ to $\mathcal{C}$. We guess the word breaking soundness of DDH with probability 1/2. Hence $\Pr[E] \leq 2 \cdot \varepsilon_s(\kappa)$ where ε_s is the soundness error of DDH.

Now (p_1, z_1) and (p_2, z_2) are secret shares of the same polynomial $f = \rho X + \mathsf{sk}_\Sigma$. Hence x is uniquely determined via

$$\mathsf{sk}_\Sigma = f(0) = z_1 \frac{p_2}{p_2 - p_1} + z_2 \frac{p_1}{p_1 - p_2}.$$

Since the key was set up honestly, we have $\Pr[S_1] = 0$ and in total $\Pr[S_0] \leq 2 \cdot \varepsilon_s(\kappa)$, which concludes the proof.

Recall that the crs of NIZK proof systems instantiated by applying the Fiat-Shamir transform to a Σ-protocol consists of a globally fixed hash function, e.g. SHA-256 or SHA-3. Consequently, this hash function can simply be part of the DAPS description, removed from the key generation and globally fixed. Now one can observe that the properties of the proof system do not require a trusted setup. So even when considering keys generated by the adversary, this observation and the perfect correctness of the encryption scheme ensure that our DAPS construction guarantees the successful extraction of the signing key of the underlying signature scheme. We now give a sketch of the proof.

Theorem 3. *If the NIZK proof system is sound and instantiated by applying the Fiat-Shamir transform to the Σ-protocol in Scheme 2, then* $\Sigma\text{-DAPS}$ *provides* wDSE^* *security.*

Proof (Sketch). We observe that the only parameter which needs to be controlled by the simulator in the proof of Theorem 2 is the crs. Now, since there is no crs in Fiat-Shamir transformed Σ-protocols, wDSE^* follows from this property, Transition $0 \rightarrow 1$ of Theorem 2, and the observation that sk_Σ is then uniquely determined by the two shares included in the signatures.

6.7 DAPS from ECDSA

As an example we give a concrete instantiation of our DAPS construction based on ECDSA, dubbed ECDSA-DAPS. The full scheme is presented in Scheme 4. Furthermore, we state the following corollaries.

Corollary 1. *If ECDSA is* EUF-CMA*-secure, and the NIZK proof system is adaptive zero-knowledge, then* ECDSA-DAPS *is an* EUF-CMA*-secure DAPS in the random oracle model.*

Corollary 2. *If the NIZK proof system is sound and instantiated by applying the Fiat-Shamir transform to the Σ-protocol in Scheme 2, then* ECDSA-DAPS *provides* wDSE^* *security.*

The two corollaries follow directly from the observation that ECDSA is included in the class C and Theorem 1, and Theorem 3.

6.8 Further DAPS

Our technique to construct DAPS can also be applied to the Schnorr signature scheme (cf. Appendix B) and the finite-field variant DSA. In particular, the latter is straightforward given the construction of ECDSA-DAPS in Scheme 4 and for brevity we omit the scheme. Besides DSA and Schnorr, EdDSA [BDL+12] also belongs to the class C of signatures schemes and can be extended to a DAPS in the same way. Consequently, our DAPS construction can easily be instantiated with EdDSA and curves `ed25519` [Ber06] or `ed448` [Ham15]. Even more generally, our approach towards DAPS can generically be applied to any signature schemes in the discrete logarithm setting from class C. Straightforwardly, if the public key is a single group element and otherwise for any scheme having public keys $k > 1$ group elements one simply has to combine the signature scheme with k copies of our technique. Our approach might also be applied beyond discrete logarithm based schemes if the respective setting provides a suitable encryption scheme, verifiable secret sharing scheme for secret keys and a non-interactive proof system.

Scheme	Sign	Verify	$\lvert\mathsf{pk}\rvert$	$\lvert\sigma\rvert$	**Setting**	**Model**
PS	$\ell\,\mathrm{E}_k^k$	$l\,\mathrm{E}_k^k$	k	ℓk	F	ROM
H2[GQ]	$2\,\mathrm{E}_{k/2}^{k/2}+\mathrm{E}_k^\ell$	E_k^ℓ	$3k$	$k+\ell$	F	ROM
ID2[GQ]	$4\,\mathrm{E}_{k/2}^{k/2}+2\,\mathrm{E}_k^\ell$	$3\,\mathrm{E}_k^\ell$	$3k$	$k+1$	F	ROM
H2[MR]	$2\,\mathrm{E}_{k/2}^{k/2}+\mathrm{E}_k^\ell$	$\frac{2\ell}{3}\,\mathrm{M}_k$	k	$k+\ell$	F	ROM
RKS	$(r-1)h\,\mathrm{S}_{\mathbb{G}}$	$2h\,\mathrm{S}_{\mathbb{G}}$	$2s_{\mathbb{G}}+k$	$((h-1)r+1)s_{\mathbb{G}}$	DL	ROM
Σ-DAPS	$\mathsf{Sign}_\Sigma+4\,\mathrm{S}_{\mathbb{G}}$	$\mathsf{Verify}_\Sigma+6\,\mathrm{S}_{\mathbb{G}}$	$\lvert\mathsf{pk}_\Sigma\rvert+(1+2n)s_{\mathbb{G}}$	$\lvert\sigma_\Sigma\rvert+3s_{\mathbb{Z}_q}$	DL	ROM
ECDSA-DAPS	$5\,\mathrm{S}_{\mathbb{G}}$	$8\,\mathrm{S}_{\mathbb{G}}$	$(2+2n)s_{\mathbb{G}}$	$5s_{\mathbb{Z}_q}$	DL	ROM
ECDSA	$\mathrm{S}_{\mathbb{G}}$	$2\,\mathrm{S}_{\mathbb{G}}$	$s_{\mathbb{G}}$	$2s_{\mathbb{Z}_q}$	DL	ROM

Table 2: Operation count, sizes of public keys (pk) and signatures (σ). Factoring-based: $\mathrm{E}_m^{m'}$ exponentiation with modulus of size m and exponent of size m', M_m multiplication with modulus of size m, k size of modulus, ℓ size of hash digest. DL-based: $\mathrm{S}_{\mathbb{G}}$ scalar multiplication and $s_{\mathbb{G}}$ size of an element in group $\mathbb{G}$, n number of addresses. RKS: r arity and h height of the tree, k size of PRF output.

```
$\underline{\mathsf{KGen}_\mathsf{D}(1^\kappa, n)}$: Let $\mathcal{G} = (\mathbb{G}, q, g)$ and $H : \{0,1\}^* \to \mathbb{Z}_q$ be a hash function mapping
    exactly to the order of the group. Let $\mathsf{sk}_\Sigma \xleftarrow{R} \mathbb{Z}_q^*$ and $x_E \xleftarrow{R} \mathbb{Z}_q^*$, and set $\mathsf{pk}_\Sigma \leftarrow$
    $g^{\mathsf{sk}_\Sigma}$ and $\mathsf{pk}_E \leftarrow g^{x_E}$. Let $(\rho_i)_{i\in[n]} \xleftarrow{R} (\mathbb{Z}_q^*)^n$ and $(r_i)_{i\in[n]} \xleftarrow{R} (\mathbb{Z}_q^*)^n$. Set $(C_i)_{i\in[n]} \leftarrow$
    $(g^{r_i}, \mathsf{pk}_E^{r_i} g^{\rho_i})_{i\in[n]}$. Let $\mathsf{crs} \leftarrow \mathsf{Setup}_\mathsf{DDH}(1^\kappa)$. Let $\mathsf{sk} \leftarrow (\mathsf{sk}_\Sigma, (r_i, \rho_i)_{i\in[n]})$ and $\mathsf{pk} \leftarrow$
    $(\mathsf{pk}_\Sigma, \mathsf{pk}_E, (C_i)_{i\in[n]}, \mathsf{crs})$ and return $(\mathsf{sk}, \mathsf{pk})$.
$\underline{\mathsf{Sign}_\mathsf{D}(\mathsf{sk}, m)}$: Parse $\mathsf{sk}$ as $(\mathsf{sk}_\Sigma, (r_i, \rho_i)_{i\in[n]})$. Parse $m$ as $(i, p)$ with $i \leq n$ and $p \in \mathbb{Z}_q^*$.
     1. Choose $k \xleftarrow{R} \mathbb{Z}_q^*$
     2. compute $R \leftarrow g^k$
     3. let $r \leftarrow R_x \pmod{q}$ and if $r = 0$ goto step 1
     4. let $s \leftarrow k^{-1}(H(m) + r\mathsf{sk}_\Sigma) \pmod{q}$ and if $s = 0$ goto step 1
     5. let $z \leftarrow \rho_i p + \mathsf{sk}_\Sigma$
     6. let $C_2' \leftarrow C_{i,2} \cdot (\mathsf{pk}_\Sigma \cdot g^{-z})^{\frac{1}{p}}$
     7. $\pi \leftarrow \mathsf{Proof}_\mathsf{DDH}(\mathsf{crs}, (g, \mathsf{pk}_E, C_{i,1}, C_2'), r_i)$
     8. return $(r, s, z, \pi)$
$\underline{\mathsf{Verify}_\mathsf{D}(\mathsf{pk}, m, \sigma)}$: Parse $\mathsf{pk}$ as $(\mathsf{pk}_\Sigma, \mathsf{pk}_E, (C_i)_{i\in[n]}, \mathsf{crs}, \cdot)$, $m$ as $(i, p)$ with $i \leq n$, and $\sigma$
    as $(r, s, z, \pi)$.
     1. If $r = 0 \ \vee \ s = 0$ return 0
     2. let $z \leftarrow H(m)$ and $w \leftarrow s^{-1} \pmod{q}$
     3. let $u_1 \leftarrow zw \pmod{q}$ and $u_2 \leftarrow rw \pmod{q}$
     4. let $R \leftarrow g^{u_1} \cdot \mathsf{pk}_\Sigma^{u_2}$
     5. if $R_x = r \pmod{q}$ return 1 and return 0 otherwise
     6. let $C_2' \leftarrow C_{i,2} \cdot (\mathsf{pk}_\Sigma \cdot g^{-z})^{\frac{1}{p}}$
     7. return $\mathsf{Verify}_\mathsf{DDH}(\mathsf{crs}, (g, \mathsf{pk}_E, C_{i,1}, C_2'), \pi)$
$\underline{\mathsf{Ex}_\mathsf{D}(\mathsf{pk}, m_1, m_2, \sigma_1, \sigma_2)}$: Parse $\sigma_i$ as $(\cdot, z_i, \cdot)$, $m_i$ as $(a_i, p_i)$ and $\mathsf{pk}$ as $(\cdot, \cdot, \cdot, \cdot)$.
     1. If $m_1$ and $m_2$ are not colliding, return $\perp$
     2. if $\mathsf{Verify}_\mathsf{D}(\mathsf{pk}, m_i, \sigma_i) = 0$ for any $i$, return $\perp$
     3. let $\mathsf{sk}_\Sigma \leftarrow z_1 \frac{p_2}{p_2 - p_1} + z_2 \frac{p_1}{p_1 - p_2}$
     4. return $\mathsf{sk}_\Sigma$
```

Scheme 4: ECDSA-DAPS**: DAPS from ECDSA.**

6.9 *N*-Times-Authentication-Preventing Signatures

Finally, we observe that our techniques can easily be generalized to what we call N-times-authentication-preventing signatures (NAPS). That is, signature schemes where creating N signatures with respect to the same address leaks the secret key while they are unforgeable as long as there are $< N$ signatures for every address. While an extension of the formal model is straightforward and therefore omitted, we subsequently sketch the construction.

Essentially, instead of computing z by evaluating a degree 1 polynomial $f_i(X) = \rho_i X + \mathsf{sk}_\Sigma \in \mathbb{Z}_q[X]$ associated to address i at the payload p, we can generalize our approach to a degree $N-1$ polynomial $f_i(X) = \mathsf{sk}_\Sigma + \sum_{j\in[N-1]} \rho_{ij} X^j \in \mathbb{Z}_q[X]$. The evaluation in the encrypted domain works likewise (when including the values ρ_{ij} in encrypted form in the public key) and the proof Π remains the same. Also the signature size is not influenced by this extension. Finally, the proofs easily generalize from 2 to N and hold under exactly the same argumentation. Thus we do not restate them.

6.10 Comparison with Previous Work

Now we want to compare the existing instantiations of DAPS in the factoring (F) and discrete logarithm (DL) setting with the ones presented in this paper. We stress that we are interested in cryptographic settings that are currently widely used and thus do not consider the lattice-based DAPS in [BKN17]. In Table 2, which is based on the recent work in [BPS17], we present a comparison of existing DAPS in terms of operation count and sizes of public keys and signatures. For reference, we also include the costs of ECDSA.

The costs of the factoring-based schemes are dominated by exponentiations with the respective RSA modulus. Observe that the savings in the signature size of one hash digest when applying the ID2 transform instead of the H2 transform, comes at the cost of twice the amount of operations during signing and thrice the operations during verification. While choosing MR as identification-scheme over GQ allows to reduce the operation count for verification and the size of the public key, signing costs are the same.

The performance of RKS largely depends on the concrete choice for the Merkle tree. When using a pseudorandom function (PRF) with k bit output, the arity of the tree r and the height h need to satisfy $r^h \geq 2^{2k}$. Additionally, the group $\mathbb{G}$ needs to be compatible with the PRF, i.e., $\log_2 |\mathbb{G}| = 2k$. For example, when using a binary tree ($r = 2$), then the height needs to be at least $2k$. While increasing the arity decreases the verification times, signing times and signature sizes increase.

When looking at our DAPS construction, the operation count of signing and verification takes an extra 4, respectively 6 group operations. The signature contains 3 additional $\mathbb{Z}_q$ elements. When instantiating our construction with ECDSA, signing requires 5 group operations in total, and verification takes 8 group operations. Signatures consists of 5 $\mathbb{Z}_q$ elements.

7 Implementation

We now present an implementation[11] of our ECDSA-DAPS based on the widely used OpenSSL[12] library and its ECDSA implementation. We note that OpenSSL's ECDSA implementation can be extended without any modifications. But also any other ECDSA implementation can be extended in the same way as long as an API for the necessary group operations is available. Note that any implementation of our DAPS construction is extendable to NAPS.

7.1 Benchmarking ECDSA-DAPS

For comparing our construction with existing DAPS implementations, we benchmarked ECDSA-DAPS using curves `secp256k1` and `prime256v1` and the DAPS

[11] The implementation is available at https://github.com/IAIK/daps-dl.

[12] https://openssl.com.

schemes H2[GQ], ID2[GQ], and H2[MR] from [BPS17] with a 2048 bit modulus. The benchmarks were performed on an Intel Core i7-4790 CPU and 16 GB RAM running Ubuntu 17.04 and the results are presented in Table 3. We omit the PS and RKS DAPS in this comparison, as they are by far not competitive; neither in terms of signature size nor performance (cf. [BPS17, Figure 21] for an overview). For reference, we also include sizes and timings for ECDSA. For the sizes required to store elliptic curve points, we assume that point compression is used.[13]

Scheme	Sign [ms]	Verify [ms]	$\|\mathsf{sk}\|$ [bits]	$\|\mathsf{pk}\|$ [bits]	$\|\sigma\|$ [bits]
H2[GQ]	1.12	0.65	4096	6144	2304
ID2[GQ]	2.12	2.06	4096	6144	2049
H2[MR]	1.36	0.58	4096	2048	2304
ECDSA-DAPS (s)	0.76	1.33	$256 \cdot (1 + 2n)$	$514 \cdot (1 + n)$	1280
ECDSA-DAPS (p)	0.23	0.35	$256 \cdot (1 + 2n)$	$514 \cdot (1 + n)$	1280
ECDSA (s)	0.09	0.35	256	257	512
ECDSA (p)	0.06	0.21	256	257	512

Table 3: Timings and sizes of private keys (sk), public keys (pk) and signatures (σ) with n addresses. The curves secp256k1 and prime256v1 are denoted as s and p, respectively.

Compared to H2[GQ], ID2[GQ], and H2[MR], ECDSA-DAPS using the curve prime256v1 is an order of magnitude faster when signing and verification is of the same order of magnitude, yet slightly faster as the faster H2 schemes. For ECDSA-DAPS using secp256k1 the picture for verification is slightly different: verification is comparable to the slower ID2[GQ] scheme. The difference in the signing and verifications times that can be observed in conventional ECDSA and ECDSA-DAPS when switching curves, and it shows that OpenSSL includes a more optimized implementation of the arithmetic on prime256v1.

8 Conclusion

In this paper we asked whether one can construct DAPS from signature schemes used in practice. We affirmatively have answered this question by presenting provably secure DAPS schemes, among others, from the widely used ECDSA signature scheme. They are the shortest among all existing DAPS schemes and improve over the most efficient factoring and discrete logarithm based schemes. Moreover, we showed how to extend our approach to N-times-authentication-preventing signatures for any $N > 2$. We provided an integration into the

[13] We store the x-coordinate and a bit indicating the "sign" of the y-coordinate. So points require $b + 1$ bits instead of $2b$ bits for b-bit curves.

OpenSSL library to foster fast adoption in practical applications, of which we discuss some interesting ones in this paper.

Acknowledgements. The authors have been supported by EU H2020 Project PRISMACLOUD, grant agreement n°644962. We thank Bertram Poettering for his comments and in particular for pointing out an issue in the proof of Theorem 1 in a previous version.

References

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.

[BDL⁺12] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. *J. Cryptographic Engineering*, 2(2):77–89, 2012.

[Ber06] Daniel J. Bernstein. Curve25519: New diffie-hellman speed records. In *Public Key Cryptography*, volume 3958 of *Lecture Notes in Computer Science*, pages 207–228. Springer, 2006.

[BKN17] Dan Boneh, Sam Kim, and Valeria Nikolaenko. Lattice-based DAPS and generalizations: Self-enforcement in signature schemes. In *ACNS*, volume 10355 of *Lecture Notes in Computer Science*, pages 457–477. Springer, 2017.

[BPS16] Mihir Bellare, Bertram Poettering, and Douglas Stebila. From identification to signatures, tightly: A framework and generic transforms. In *ASIACRYPT (2)*, volume 10032 of *Lecture Notes in Computer Science*, pages 435–464, 2016.

[BPS17] Mihir Bellare, Bertram Poettering, and Douglas Stebila. Deterring certificate subversion: Efficient double-authentication-preventing signatures. In *Public Key Cryptography (2)*, volume 10175 of *Lecture Notes in Computer Science*, pages 121–151. Springer, 2017.

[Bro02] Daniel R. L. Brown. Generic groups, collision resistance, and ECDSA. *IACR Cryptology ePrint Archive*, 2002:26, 2002.

[Bro05] Daniel R. L. Brown. Generic groups, collision resistance, and ECDSA. *Des. Codes Cryptography*, 35(1):119–152, 2005.

[CFN88] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In *CRYPTO*, volume 403 of *Lecture Notes in Computer Science*, pages 319–327. Springer, 1988.

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer, 2001.

[CP92] David Chaum and Torben P. Pedersen. Wallet databases with observers. In *CRYPTO*, volume 740 of *Lecture Notes in Computer Science*, pages 89–105. Springer, 1992.

[DAM⁺15] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. A search engine backed by internet-wide scanning. In *ACM Conference on Computer and Communications Security*, pages 542–553. ACM, 2015.

[DY05] Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In *Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 416–431. Springer, 2005.

[Fel87] Paul Feldman. A practical scheme for non-interactive verifiable secret sharing. In *FOCS*, pages 427–437. IEEE Computer Society, 1987.

[FKMV12] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the non-malleability of the fiat-shamir transform. In *INDOCRYPT*, volume 7668 of *Lecture Notes in Computer Science*, pages 60–79. Springer, 2012.

[FKP16] Manuel Fersch, Eike Kiltz, and Bertram Poettering. On the provable security of (EC)DSA signatures. In *ACM Conference on Computer and Communications Security*, pages 1651–1662. ACM, 2016.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[Gam84] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer, 1984.

[GQ88] Louis C. Guillou and Jean-Jacques Quisquater. A ”paradoxical” indentity-based signature scheme resulting from zero-knowledge. In *CRYPTO*, volume 403 of *Lecture Notes in Computer Science*, pages 216–231. Springer, 1988.

[Ham15] Mike Hamburg. Ed448-goldilocks, a new elliptic curve. *IACR Cryptology ePrint Archive*, 2015:625, 2015.

[HPM94] Patrick Horster, Holger Petersen, and Markus Michels. Meta-elgamal signature schemes. In *ACM Conference on Computer and Communications Security*, pages 96–107. ACM, 1994.

[KAC12] Ghassan Karame, Elli Androulaki, and Srdjan Capkun. Double-spending fast payments in bitcoin. In *ACM Conference on Computer and Communications Security*, pages 906–917. ACM, 2012.

[KAR$^+$15] Ghassan O. Karame, Elli Androulaki, Marc Roeschlin, Arthur Gervais, and Srdjan Capkun. Misbehavior in bitcoin: A study of double-spending and accountability. *ACM Trans. Inf. Syst. Secur.*, 18(1):2:1–2:32, 2015.

[KMP16] Eike Kiltz, Daniel Masny, and Jiaxin Pan. Optimal security proofs for signatures from identification schemes. In *CRYPTO (2)*, volume 9815 of *Lecture Notes in Computer Science*, pages 33–61. Springer, 2016.

[KR00] Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *NDSS*. The Internet Society, 2000.

[MR02] Silvio Micali and Leonid Reyzin. Improving the exact security of digital signature schemes. *J. Cryptology*, 15(1):1–18, 2002.

[MRV99] Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In *FOCS*, pages 120–130. IEEE Computer Society, 1999.

[MS02] John Malone-Lee and Nigel P. Smart. Modifications of ECDSA. In *Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2002.

[Poe18] Bertram Poettering. Shorter double-authentication preventing signatures for small address spaces. In *AFRICACRYPT*, volume 10831 of *Lecture Notes in Computer Science*, pages 344–361. Springer, 2018.

[Por13] Thomas Pornin. Deterministic usage of the digital signature algorithm (DSA) and elliptic curve digital signature algorithm (ECDSA). *RFC*, 6979:1–79, 2013.

[PS96] David Pointcheval and Jacques Stern. Security proofs for signature schemes. In *EUROCRYPT*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398. Springer, 1996.

[PS14] Bertram Poettering and Douglas Stebila. Double-authentication-preventing signatures. In *ESORICS (1)*, volume 8712 of *Lecture Notes in Computer Science*, pages 436–453. Springer, 2014.

[PS17] Bertram Poettering and Douglas Stebila. Double-authentication-preventing signatures. *Int. J. Inf. Sec.*, 16(1):1–22, 2017.

[PWH$^+$17] Dimitrios Papadopoulos, Duane Wessels, Shumon Huque, Moni Naor, Jan Vcelák, Leonid Reyzin, and Sharon Goldberg. Can NSEC5 be practical for DNSSEC deployments? *IACR Cryptology ePrint Archive*, 2017:99, 2017.

[RKS15] Tim Ruffing, Aniket Kate, and Dominique Schröder. Liar, liar, coins on fire!: Penalizing equivocation by loss of bitcoins. In *ACM Conference on Computer and Communications Security*, pages 219–230. ACM, 2015.

[Sch89] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 1989.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer, 1997.

[SW17] Haya Shulman and Michael Waidner. One key to sign them all considered vulnerable: Evaluation of DNSSEC in the internet. In *NSDI*, pages 131–144. USENIX Association, 2017.

[vRJS16] Roland van Rijswijk-Deij, Mattijs Jonker, and Anna Sperotto. On the adoption of the elliptic curve digital signature algorithm (ECDSA) in DNSSEC. In *CNSM*, pages 258–262. IEEE, 2016.

A Cryptographic Assumptions

Subsequently, we present the decisional Diffie-Hellman (DDH or 1-LIN) and decision linear (DLIN or 2-LIN) assumptions, very common assumptions underlying the IND-CPA security of versions of the ElGamal encryption scheme.

Definition 10 (DDH). *The DDH assumptions holds relative to $\mathcal{G} = (\mathbb{G}, q, g)$, if for all PPT adversaries $\mathcal{A}$, there is a negligible function ε such that*

$$\left| \Pr \left[\begin{array}{l} x, y, z \xleftarrow{R} \mathbb{Z}_q, \\ b^* \leftarrow \mathcal{A}\left(g^x, g^y, g^{b \cdot xy + (1-b)z}\right) \end{array} : b = b^* \right] - \frac{1}{2} \right| \leq \varepsilon(\kappa)$$

Definition 11 (DLIN). *The DLIN assumptions holds relative to $\mathcal{G} = (\mathbb{G}, q, g)$, if for all PPT adversaries $\mathcal{A}$, there is a negligible function ε such that*

$$\left| \Pr \left[\begin{array}{l} u, v, h \xleftarrow{R} \mathbb{G},\ x, y, z \xleftarrow{R} \mathbb{Z}_q, \\ b^* \leftarrow \mathcal{A}\left(\begin{array}{l} u, v, h, u^x, v^y, \\ h^{b \cdot (x+y) + (1-b)z} \end{array} \right) \end{array} : b = b^* \right] - \frac{1}{2} \right| \leq \varepsilon(\kappa)$$

B Schnorr Signature Scheme

The Schnorr signature scheme [Sch89] can be seen as a prime example of a signature scheme obtained from an identification scheme using the Fiat-Shamir heuristic [FS86]. We present an instantiation of Schnorr in Scheme 5. The Schnorr signature scheme can be shown to provide EUF-CMA security in the random oracle model (ROM) under the DLP in $\mathbb{G}$ by using the now popular rewinding technique [PS96] (cf. also [KMP16] for a recent treatment on tightness and optimality of such reductions).

$\underline{\mathsf{KGen}_{\mathsf{Schnorr}}(1^\kappa)}$: Let $\mathcal{G} = (\mathbb{G}, q, g)$. Choose $x \overset{R}{\leftarrow} \mathbb{Z}_q^*$ and set $\mathsf{sk} \leftarrow x$ and $\mathsf{pk} \leftarrow g^x$ and return $(\mathsf{sk}, \mathsf{pk})$.

$\underline{\mathsf{Sign}_{\mathsf{Schnorr}}(\mathsf{sk}, m)}$: Parse sk as x and choose $k \overset{R}{\leftarrow} \mathbb{Z}_q^*$. Compute $c \leftarrow H(g^k \| m)$, $s \leftarrow k - cx$ and return (c, s).

$\underline{\mathsf{Verify}_{\mathsf{Schnorr}}(\mathsf{pk}, m, \sigma)}$: Parse σ as (c, s) and compute $r \leftarrow g^s \mathsf{pk}^c$. Return 1 if $c = H(r \| m)$ and 0 otherwise.

Scheme 5: Schnorr signature scheme.

C DSE* Security of DAPS

We recall the DSE* security notion of DAPS. The game is depicted in Figure 6, where in contrast to Figure 3 the keys are allowed to be generated by the adversary.

Definition 12 **(DSE* [PS14]).** *A DAPS scheme provides double-signature extraction (DSE*), if for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that*

$$\Pr\left[\mathbf{Exp}^{\mathsf{DSE}^*}_{\mathcal{A},\mathsf{DAPS}^*}(\kappa) = 1\right] \leq \varepsilon(\kappa),$$

where the corresponding experiment is depicted in Figure 6.

```
$\mathbf{Exp}^{\mathsf{DSE}^*}_{\mathcal{A},\mathsf{DAPS}}(\kappa)$:
    $(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2) \leftarrow \mathcal{A}(1^\kappa)$
    return 0, if $m_1$ and $m_2$ are not colliding
    $v_i \leftarrow \mathsf{Verify_D}(\mathsf{pk_D}, m_i, \sigma_i)$ for $i \in [2]$
    return 0, if $v_1 = 0$ or $v_2 = 0$
    $\mathsf{sk}'_\mathsf{D} \leftarrow \mathsf{Ex_D}(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2)$
    return 1, if $\mathsf{sk}'$ is not the secret key corresponding to $\mathsf{pk_D}$
    return 0
```

Fig. 6: DSE* **security for** DAPS.

D IND-CPA Security

$\mathsf{IND\text{-}CPA}$ security of an encryption scheme Ω is depicted in Figure 7.

Definition 13 ($\mathsf{IND\text{-}CPA}$). *A public key encryption scheme Ω is* $\mathsf{IND\text{-}CPA}$ *secure, if for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that*

$$\Pr\left[\mathbf{Exp}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\Omega}(\kappa) = 1\right] \leq \varepsilon(\kappa),$$

where the corresponding experiment is depicted in Figure 7.

```
$\mathbf{Exp}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\Omega}(\kappa)$
  $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\kappa)$
  $b \leftarrow \{0, 1\}$
  $(m_0, m_1, \mathtt{state}_\mathcal{A}) \leftarrow \mathcal{A}(\mathsf{pk})$
  if $m_0 \notin \mathcal{M} \vee m_1 \notin \mathcal{M}$, let $C \leftarrow \bot$
  else, let $C^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$
  $b^* \leftarrow \mathcal{A}(C^*, \mathtt{state}_\mathcal{A})$
  return 1, if $b^* = b$
  return 0
```

Fig. 7: IND-CPA security.

E Σ-Protocols

Let $L \subseteq \mathsf{X}$ be an $\mathbf{NP}$-language with associated witness relation R so that $L = \{x \mid \exists w : R(x, w) = 1\}$. A Σ-protocol for language L is defined as follows.

Definition 14. *A Σ-protocol for language L is an interactive three-move protocol between a PPT prover $\mathsf{P} = (\mathsf{Commit}, \mathsf{Prove})$ and a PPT verifier $\mathsf{V} = (\mathsf{Challenge}, \mathsf{Verify})$, where P makes the first move and transcripts are of the form $(\mathsf{a}, \mathsf{c}, \mathsf{s}) \in \mathsf{A} \times \mathsf{C} \times \mathsf{S}$. Additionally they satisfy the following properties:*

Completeness *A Σ-protocol for language L is complete, if for all security parameters κ, and for all $(x, w) \in R$, it holds that*

$$\Pr[\langle \mathsf{P}(1^\kappa, x, w), \mathsf{V}(1^\kappa, x)\rangle = 1] = 1.$$

Special Soundness *A Σ-protocol for language L is special sound, if there exists a PPT extractor $\mathcal{E}$ so that for all x, and for all sets of accepting transcripts $\{(\mathsf{a}, \mathsf{c}_i, \mathsf{s}_i)\}_{i \in [2]}$ with respect to x where $\mathsf{c}_1 \neq \mathsf{c}_2$, generated by any algorithm with polynomial runtime in κ, it holds that*

$$\Pr\left[\begin{array}{l} w \leftarrow \mathcal{E}(1^\kappa, x, \\ \{(\mathsf{a}, \mathsf{c}_i, \mathsf{s}_i)\}_{i \in [2]}) \end{array} : \ (x, w) \in R\right] \geq 1 - \varepsilon(\kappa).$$

Special Honest-Verifier Zero-Knowledge *A Σ-protocol is special honest-verifier zero-knowledge, if there exists a PPT simulator $\mathcal{S}$ so that for every $x \in L$ and every challenge* c *from the challenge space, it holds that a transcript* $(\mathsf{a}, \mathsf{c}, \mathsf{s})$*, where* $(\mathsf{a}, \mathsf{s}) \leftarrow \mathcal{S}(1^\kappa, x, \mathsf{c})$ *is indistinguishable from a transcript resulting from an honest execution of the protocol.*

F NIZK Security Properties

Definition 15 (Completeness). *A non-interactive proof system for language L is complete, if for all $\kappa \in \mathbb{N}$, for all* $\mathsf{crs} \leftarrow \mathsf{Setup}_\Pi(1^\kappa)$*, for all $x \in L$, for all w such that $R(x, w) = 1$, and for all* $\pi \leftarrow \mathsf{Proof}_\Pi(\mathsf{crs}, x, w)$*, we have that* $\mathsf{Verify}_\Pi(\mathsf{crs}, x, \pi) = 1$.

This captures perfect completeness.

Definition 16 (Soundness). *A non-interactive proof system for language L is sound, if for every PPT adversary $\mathcal{A}$ there exists a negligible function ε such that:*

$$\Pr\begin{bmatrix} \mathsf{crs} \leftarrow \mathsf{Setup}_\Pi(1^\kappa), & \mathsf{Verify}_\Pi(\mathsf{crs}, x, \pi) \\ (x, \pi) \leftarrow \mathcal{A}(\mathsf{crs}) & = 1 \wedge x \notin L \end{bmatrix} \leq \varepsilon(\kappa).$$

Definition 17 (Zero-Knowledge). *A non-interactive proof system for language L is zero-knowledge, if there exists an efficient simulator* $\mathsf{S} = (\mathcal{S}_1, \mathcal{S}_2)$ *such that for any efficient adversary $\mathcal{A}$ there exist a negligible function $\varepsilon_1(\cdot)$ such that:*

$$\left| \begin{matrix} \Pr[\mathsf{crs} \leftarrow \mathsf{Setup}_\Pi(1^\kappa) : \mathcal{A}(\mathsf{crs}) = 1] & - \\ \Pr[(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^\kappa) : \mathcal{A}(\mathsf{crs}) = 1] & \end{matrix} \right| \leq \varepsilon_1(\kappa),$$

and for any efficient adversary $\mathcal{A}$ there exists a negligible function $\varepsilon_2(\cdot)$ such that

$$\left| \Pr[\mathsf{Zero\text{-}Knowledge}^\Pi_{\mathcal{A},\mathsf{S}}(\kappa) = 1] - \frac{1}{2} \right| \leq \varepsilon_2(\kappa),$$

where the corresponding experiment is depicted in Figure 8.

```
**Experiment** $\mathsf{Zero\text{-}Knowledge}^{\Pi}_{\mathcal{A},\mathsf{S}}(\kappa)$
  $b \leftarrow \{0,1\}$
  $(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^{\kappa})$
  $b^* \leftarrow \mathcal{A}^{\mathsf{P}_b(\cdot,\cdot)}(\mathsf{crs})$
    where oracle $\mathsf{P}_0$ on input $(x, w)$:
      return $\pi \leftarrow \mathsf{Proof}_{\Pi}(\mathsf{crs}, x, w)$, if $(x, w) \in R$
      return $\perp$
    and oracle $\mathsf{P}_1$ on input $(x, w)$:
      return $\pi \leftarrow \mathcal{S}_2(\mathsf{crs}, \tau, x)$, if $(x, w) \in R$
      return $\perp$
  return 1, if $b = b^*$
  return 0
```

Fig. 8: Zero-Knowledge

5

Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives

Publication Data

Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1825–1842. ACM, 2017. URL: https://doi.org/10.1145/3133956.3133997

The appended paper is an author-created full version available at https://eprint.iacr.org/2017/279 and was relayouted to fit the formatting of this thesis.

Contributions

- This paper is a merge of [GCZ16] and [DOR+16]. The author is one of the main authors of the latter.
- The author contributed to the selection of symmetric primitives, the implementation and the comparison with other post-quantum secure signature schemes.

Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives*

Melissa Chase[1], David Derler[2], Steven Goldfeder[3], Claudio Orlandi[4], Sebastian Ramacher[2], Christian Rechberger[2,5], Daniel Slamanig[6], and Greg Zaverucha[1]

[1] Microsoft Research
[2] Graz University of Technology
[3] Princeton University
[4] Aahrus University
[5] Denmark Technical University
[6] AIT Austrian Institute of Technology

Abstract. We propose a new class of *post-quantum* digital signature schemes that: (a) derive their security entirely from the security of symmetric-key primitives, believed to be quantum-secure, and (b) have extremely small keypairs, and, (c) are highly parameterizable.

In our signature constructions, the public key is an image $y = f(x)$ of a one-way function f and secret key x. A signature is a non-interactive zero-knowledge proof of x, that incorporates a message to be signed. For this proof, we leverage recent progress of Giacomelli et al. (USENIX'16) in constructing an efficient Σ-protocol for statements over general circuits. We improve this Σ-protocol to reduce proof sizes by a factor of two, at no additional computational cost. While this is of independent interest as it yields more compact proofs for any circuit, it also decreases our signature sizes.

We consider two possibilities to make the proof non-interactive: the Fiat-Shamir transform and Unruh's transform (EUROCRYPT'12, '15,'16). The former has smaller signatures, while the latter has a security analysis in the quantum-accessible random oracle model. By customizing Unruh's transform to our application, the overhead is reduced to 1.6x when compared to the Fiat-Shamir transform, which does not have a rigorous post-quantum security analysis.

We implement and benchmark both approaches and explore the possible choice of f, taking advantage of the recent trend to strive for practical symmetric ciphers with a particularly low number of multiplications and end up using LowMC (EUROCRYPT'15).

Keywords: Post-quantum cryptography, zero-knowledge, signatures, block cipher, Fiat-Shamir, Unruh, implementation

* The performance figures presented here are somewhat outdated. For up to date figures see https://microsoft.github.io/Picnic/. This is the full version of a paper which appears in CCS'17: 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017. ACM, New York, NY, USA. This paper is a merge of [34,46].

1 Introduction

More than two decades ago Shor published his polynomial-time quantum algorithm for factoring and computing discrete logarithms [81]. Since then, we know that a sufficiently powerful quantum computer is able to break nearly all public key cryptography used in practice today. This motivates the invention of cryptographic schemes with *post quantum* (PQ) security, i.e., security against attacks by a quantum computer. Even though no sufficiently powerful quantum computer currently exists, NIST recently announced a post-quantum crypto project[7] to avoid a rushed transition from current cryptographic algorithms to PQ secure algorithms. The project is seeking proposals for public key encryption, key exchange and digital signatures thought to have PQ security. The deadline for proposals is fall 2017.

In this paper we are concerned with constructing signature schemes for the post-quantum era. The building blocks of our schemes are interactive honest-verifier zero-knowledge proof systems (Σ-protocols) for statements over general circuits and symmetric-key primitives, that are conjectured to remain secure in a post-quantum world.

Post-Quantum Signatures. Perhaps the oldest signature scheme with post-quantum security are one-time Lamport signatures [63], built using hash functions. As Grover's quantum search algorithm can invert any black-box function [52] with a quadratic speed-up over classical algorithms, one has to double the bit size of the hash function's domain, but still requires no additional assumptions to provably achieve post-quantum security. Combined with Merkle-trees, this approach yields stateful signatures for any polynomial number of messages [71], where the state ensures that a one-time signature key from the tree is not reused. By making the tree very large, and randomly selecting a key from it (cf. [47]), along with other optimizations, yields practical stateless hash-based signatures [17].

There are also existing schemes that make structured (or number-theoretic) assumptions. Code-based signature schemes can be obtained from identification schemes based on the syndrome decoding (SD) problem [70,82,86] by applying a variant of the well-known Fiat-Shamir (FS) transform [40]. Lattice-based signature schemes secure under the short integer solution (SIS) problem on lattices following the Full-Domain-Hash (FDH) paradigm [13] have been introduced in [43]. More efficient approaches [7,9,65,66] rely on the FS transform instead of FDH. BLISS [36], a very practical scheme, also relies on the FS transform, but buys efficiency at the cost of more pragmatic assumptions, i.e., a ring version of the SIS problem. For signatures based on problems related to multivariate systems of quadratic equations only recently provably secure variants relying on the FS transform have been proposed [56].

When it comes to confidence in the underlying assumptions, hash-based signatures are arguably the preferred candidate among all existing approaches. All

[7] `http://csrc.nist.gov/groups/ST/post-quantum-crypto/`

other practical signatures require an additional structured assumption (in addition to assumptions related to hash functions).

1.1 Contributions

We contribute a novel class of practical post-quantum signature schemes. Our approach only requires symmetric key primitives like hash functions and block ciphers and *does not* require additional structured hardness assumptions.

Along the way to building our signature schemes, we make several contributions of general interest to zero-knowledge proofs both in the classical and post-quantum setting:

- We improve ZKBoo [44], a recent Σ-protocol for proving statements over general circuits. We reduce the transcript size by more than half without increasing the computational cost. We call the improved protocol ZKB++. This improvement is of general interest outside of our application to post-quantum signatures as it yields significantly more concise zero knowledge proofs even in the classical setting.
- We also show how to apply Unruh's generic transform [83,84,85] to obtain a non-interactive counterpart of ZKB++ that is secure in the quantum-accessible random oracle model (QROM; see [18]). To our knowledge, we are the first to apply Unruh's transform in an efficient signature scheme.
- Unruh's construction is generic, and does not immediately yield compact proofs. However, we specialize the construction to our application, and we find the overhead was surprisingly low – whereas a generic application of Unruh's transform incurs a 4x increase in size when compared to FS, we were able to reduce the size overhead of Unruh's transform to only 1.6x. Again, this has applications wider than our signature schemes as the protocol can be used for non-interactive post-quantum zero knowledge proofs secure in the QROM.

We build upon these results to achieve our central contribution: two concrete signature schemes. In both schemes the public key is set up to be an image $y = f(k)$ with respect to one-way function f and secret key k. We then turn an instance of ZKB++ to prove knowledge of k into two signature schemes – one using the FS transform and the other using Unruh's transform. The FS variant, dubbed Fish, yields a signature scheme that is secure in the ROM, whereas the Unruh variant, dubbed Picnic, yields a signature scheme that is secure in the QROM, and we include a complete security proof.

We review symmetric-key primitives with respect to their suitability to serve as f in our application and conclude that the LowMC family of block ciphers [4,6] is well suited. We explore the parameter space of LowMC and show that we can obtain various trade-offs between signature size and computation time. Thereby, our approach turns out to be very flexible as besides the aforementioned trade-offs we are also able to adjust the security parameter of our construction in a very fine-grained way.

We provide an implementation of both schemes for 128-bit post-quantum security, demonstrating the practical relevance of our approach. In particular, we provide two reference implementations on GitHub[8,9]. Moreover, we rigorously compare our schemes with other practical provably secure post-quantum schemes.

1.2 Related Work

We now give a brief overview of other candidate schemes and defer a detailed comparison of parameters and performance to Section 7. We start with the only existing instantiation that solely relies on standard assumptions, i.e., comes with a security proof in the standard model (SM). The remaining existing schemes rely on structured assumptions related to codes, lattices and multivariate systems of quadratic equations that are assumed to be quantum safe and have a security proof in the ROM. At the end of the section, we review the state of the art in zero-knowledge proofs for non-algebraic statements.

Hash-Based Signatures (SM). Hash-based signatures are attractive as they can be proven secure in the standard model (i.e., without ROs) under well-known properties of hash functions such as second pre-image resistance. Unfortunately, highly efficient schemes like XMSS [22] are stateful, which seems to be problematic for practical applications [68]. Stateless schemes like SPHINCS [17] are thus more desirable, but this comes at reduced efficiency and increased signature sizes. SPHINCS has a tight security reduction to security of its building blocks, i.e., hash functions, PRGs and PRFs. At the 128-bit post-quantum security level, signatures are about 41 kB in size, and keys are of size about 1 kB each.

Code-Based Signatures (ROM). In the code-based setting the most prominent and provably secure approach is to convert identification schemes due to Stern [82] and Véron [86] to signatures using FS. For the 128-bit PQ security level one obtains signature sizes of around $\approx$ 129 kB (in the best case) and public key size of $\approx$ 160 bytes.[10] We note that there are also other code-based signatures [27] based on the Niederreiter [72] dual of the McEliece cryptosystem [67], which do not come with a security reduction, have shown to be insecure [38] and also do not seem practical [64]. There is a more recent provably secure approach [37], however, it is not immediate if this leads to efficient signatures.

Lattice-Based Signatures (ROM). For lattice based signatures there are two major directions. The first are schemes that rely on the hardness of worst-to-average-case problems in standard lattices [43,66,9,30,7]. Although they are desirable from a security point of view, they suffer from huge public keys, i.e., in the orders of a few to some 10 MBs. TESLA [7] (based upon [9,66]) improves all aspects in the performance of GPV [43], but still has keys on the order of 1

[8] `https://github.com/Microsoft/Picnic`

[9] `https://github.com/IAIK/fish-begol`

[10] The given estimates are taken from a recent talk of Nicolas Sendrier (available at `https://pqcrypto.eu.org/mini.html`), as, unfortunately, there are no free implementations available.

MB. More efficient lattice-based schemes are based on ring analogues of classical lattice problems [53,36,10,3,11] whose security is related to hardness assumptions in ideal lattices. These constructions drop key sizes to the order of a few kBs. Most notable is BLISS [36,35], which achieves performance nearly comparable to RSA. However, it must be noted, that ideal lattices have not been investigated nearly as deeply as standard lattices and thus there is less confidence in the assumptions (cf. [75]).

MQ-Based Signatures (ROM). Recently, Hülsing et al. in [56] proposed a post-quantum signature scheme (MQDSS) whose security is based on the problem of solving a multivariate system of quadratic equations. Their scheme is obtained by building upon the 5-pass (or 3-pass) identification scheme in [79] and applying the FS transform. For 128-bit post-quantum security, signature sizes are about 40 kB, public key sizes are 72 bytes and secret key sizes are 64 bytes. We note that there are other MQ-based approaches like Unbalanced Oil-and-Vinegar (UOV) variants [74] or FHEv^{-} variants (cf. [76]), having somewhat larger keys (order of kBs) but much shorter signatures. However, they have no provable security guarantees, the parameter choice seems very aggressive, there are no parameters for conservative (post-quantum) security levels, and no implementations are available.

Supersingular Isogenies (QROM). Yoo et al. in [87] proposed a post-quantum signature scheme whose security is based on supersingular isogeny problems. The scheme is obtained by building upon the identification scheme in [39] and applying the Unruh transform. For 128-bit post-quantum security, signature sizes are about 140 kB, public key sizes are 768 bytes, and secret key sizes are 49 bytes.

At the same time, Galbraith et al. [41] published a preprint containing one conceptually identical isogeny-based construction, and one based on endomorphism rings. They report improved signature sizes using a time-space trade-off and only present their improvements in terms of classical security parameters.

Zero-Knowledge for Arithmetic Circuits. Zero-knowledge (ZK) proofs [49] are a powerful tool and exist for any language in **NP** [48]. Nevertheless, practically efficient proofs were until recently only known for restricted languages covering algebraic statements in certain algebraic structures, e.g., discrete logarithms [80,28] or equations over bilinear groups [51]. Expressing any **NP** language as a combination of algebraic circuits could be done for example by expressing the relation as a circuit, however for circuits of practical interest (such as hash functions or block ciphers), this quickly becomes prohibitively expensive. Even SNARKS, where proof size can be made small (and constant) and verification is highly efficient, have very costly proofs (cf. [42,15,26] and the references therein).[11] Unfortunately, signatures require small proof computation times (efficient signing procedures), and this direction is not suitable.

Quite recently, dedicated ZK proof systems for statements expressed as Boolean circuits by Jawurek et al. [58] and statements expressed as RAM programs by

[11] Using SNARKS is reasonable in scenarios where provers are extremely powerful (such as verifiable computing [42]) or the runtime of the prover is not critical (such as Zerocash [14]).

Hu et al. [55] have been proposed. As we exclusively focus on circuits, let us take a look at [58]. They proposed using garbled circuits to obtain ZK proofs, which allow efficient proofs for statements like knowledge of x for $y = \text{SHA-256}(x)$. Unfortunately, this approach is inherently interactive and thus not suitable for the design of practical signature schemes. The very recent ZKBoo protocol due to Giacomelli et al. [44], which we build upon, for the first time, allows to construct non-interactive zero-knowledge (NIZK) proofs with performance being of interest for practical applications.

QROM vs ROM. One way of arguing security for signatures obtained via the FS heuristic in the stronger QROM is to assume that it simply holds as long as the underlying protocol and the hash function used to instantiate the random oracle (RO) are quantum-secure. However, it is known [18] that there are signature schemes secure in the ROM that are insecure in the quantum-accessible ROM (QROM), i.e., when the adversary can issue quantum queries to the RO. One central issue in this context is how to handle the rewinding of adversaries within security reductions as in the FS transform [31]. Possibilities to circumvent this issue are via history-free reductions [18] or the use of oblivious commitments within the FS transform, which is not applicable to our approach. Although many existing schemes ignore QROM security, given the general uncertainty of the capabilities of quantum adversaries, we prefer to avoid this assumption. Building upon results from Unruh [83,84,85], we achieve provable security in the QROM under reasonable assumptions.

2 Building Blocks

Below, we informally recall the notion of Σ-protocols and other standard primitives.

Sigma Protocol. A *sigma protocol* (or Σ-protocol) is a three flow protocol between a prover `Prove` and a verifier `Verify`, where transcripts have the form (r, c, s). Thereby, r and s are computed by `Prove` and c is a challenge chosen by `Verify`. Let f be a relation such that $f(x) = y$, where y is common input and x is a witness known only to `Prove`. `Verify` accepts if $\phi(y, r, c, s) = 1$ for an efficiently computable predicate ϕ. There also exists an efficient simulator, given y and a randomly chosen c, outputs a transcript (r, c, s) for y that is indistinguishable from a real run of the protocol for x, y.

n-Special Soundness. A Σ-protocol has *n-special soundness* if n transcripts $(r, c_1, s_1), \ldots, (r, c_n, s_n)$ with distinct c_i guarantee that a witness may be efficiently extracted.

Fiat-Shamir. The FS transform [40] converts a Σ-protocol into a non-interactive zero knowledge proof of knowledge. A Σ-protocol consists of a transcript (r, c, s). The corresponding non-interactive proof (r', c', s') generates r' and s' as in the interactive case, but obtains $c' \leftarrow H(r')$ instead of receiving it from the verifier. This is known to be a secure NIZK in the random oracle model against standard (non-quantum) adversaries [40].

Other Building Blocks. This paper requires other common primitives, namely one-way functions, pseudorandom generators, and commitments. We use the canonical hash-based commitment and require commitments to be hiding and binding. Definitions are given in Appendix C, where we also recall the definition of signature schemes, and existential unforgeability under chosen message attacks (EUF-CMA), which is the standard security notion for signature schemes.

3 ZKBoo and ZKB++

ZKBoo is a proof system for zero-knowledge proofs on arbitrary circuits described in [45]. We recall the protocol here, and present ZKB++, an improved version of ZKBoo with proofs that are less than half the size.

3.1 ZKBoo

While ZKBoo is presented with various possible parameter options, we present only the final version from [45] with the best parameters. Moreover, while ZKBoo presents both interactive and non-interactive protocol versions, we present only the non-interactive version since our main goal is building a signature scheme.

Overview. ZKBoo builds on the MPC-in-the-head paradigm of Ishai *et al.* [57], that we describe only informally here. The multiparty computation protocol (MPC) will implement the relation, and the input is the witness. For example, the MPC could compute $y = \text{SHA-256}(x)$ where players each have a share of x and y is public. The idea is to have the prover simulate a multiparty computation protocol "in their head", commit to the state and transcripts of all players, then have the verifier "corrupt" a random subset of the simulated players by seeing their complete state. The verifier then checks that the computation was done correctly from the perspective of the corrupted players, and if so, he has some assurance that the output is correct and the prover knows x. Iterating this for many rounds then gives the verifier high assurance.

ZKBoo generalizes the idea of [57] by replacing MPC with so-called "circuit decompositions", which do not necessarily need to satisfy the properties of an MPC protocol and therefore lead to more efficient proofs in practice. Fix the number of players to three. In particular, to prove knowledge of a witness for a relation $R := \{(x, y), \phi(x) = y\}$, we begin with a circuit that computes ϕ, and then find a suitable circuit decomposition. This contains a `Share` function (that splits the input into three shares), three functions $\texttt{Output}_{i\in\{1,2,3\}}$ (that take as input all of the input shares and some randomness and produce an output share for each of the parties), and a function `Reconstruct` (that takes as input the three output shares and reconstructs the circuit's final output). This decomposition must satisfy *correctness* and 2*-privacy* which intuitively means that revealing the views of any two players does not leak information about the witness x.

The decomposition is used to construct a proof as follows: the prover runs the computation ϕ using the decomposition and commits to the views – three views per run. Then, using the FS heuristic, the prover sends the commitments and output shares from each view to the random oracle to compute a challenge – the challenge tells the prover which two of the three views to open for each of the t runs. Because of the 2-privacy property, opening two views for each run does not leak information about the witness. The number of runs, t, is chosen to achieve negligible soundness error – i.e., intuitively it would be infeasible for the prover to cheat without getting caught in at least one of the runs. The verifier checks that (1) the output of each of the three views reconstructs to y, (2) each of the two open views were computed correctly, and (3) the challenge was computed correctly.

We now give a detailed description of the non-interactive ZKBoo protocol. Throughout this paper, when we perform arithmetic on the indices of the players, we omit the implicit mod 3 to simplify the notation.

Definition 1 ((2,3)-decomposition). *Let $f(\cdot)$ be a function that is computed by an n-gate circuit ϕ such that $f(x) = \phi(x) = y$, and let κ be the security parameter. Let k_1, k_2, and k_3 be tapes chosen uniformly at random from $\{0,1\}^\kappa$ corresponding to players P_1, P_2 and P_3, respectively. Consider the following set of functions, $\mathcal{D}$:*

$$\begin{aligned}(\mathit{view}_1^{(0)},\ \mathit{view}_2^{(0)},\ \mathit{view}_3^{(0)}) &\leftarrow \mathit{Share}(x, k_1, k_2, k_3)\\ \mathit{view}_i^{(j+1)} &\leftarrow \mathit{Update}(\mathit{view}_i^{(j)}, \mathit{view}_{i+1}^{(j)}, k_i, k_{i+1})\\ y_i &\leftarrow \mathit{Output}(\mathit{View}_i)\\ y &\leftarrow \mathit{Reconstruct}(y_1, y_2, y_3)\end{aligned}$$

such that `Share` *is a potentially randomized invertible function that takes x as input and outputs the initial view for each player containing the secret share x_i of x, i.e. $\mathsf{view}_i^{(0)} = x_i$. The function* `Update` *computes the wire values for the next gate and updates the view accordingly. The function* $\mathtt{Output}_i$ *takes as input the final view, $\mathsf{View}_i \equiv \mathsf{view}_i^{(n)}$ after all gates have been computed and outputs player P_i's* output share*, y_i.*

We require correctness and 2-privacy as informally outlined before. We defer a formal definition to Appendix A.1. The concrete decomposition used by ZKBoo is presented in Appendix A.2.

ZKBoo Complete Protocol Given a $(2,3)$-decomposition $\mathcal{D}$ for a function ϕ, the ZKBoo protocol is a Σ-protocol for languages of the form $L := \{y \mid \exists\ x : y = \phi(x)\}$. We note that this directly yields a non-interactive zero-knowledge (NIZK) proof system for the same relation using well known results. We recall the details of ZKBoo in Appendix A.

Serializing the Views. In the (2,3)-decomposition, the view is updated with the output wire value for each gate. While conceptually a player's view includes the values that they computed locally, when the view is serialized, it is sufficient to include only the wire values of the gates that require non-local computations (i.e., the binary multiplication gates). The verifier can recompute the parts of the view due to local computations, and they do not need to be serialized. Giving the verifier locally computed values does not even save any computation as the verifier will still need to recompute the values in order to check them.

In ZKBoo, the serialized view includes: (1) the input share, (2) output wire values for binary multiplication gates, and (3) the output share.

The size of a view depends on the circuit as well as the ring that it is computed over. Let $\phi : (\mathbb{Z}_{2^\ell})^m \to (\mathbb{Z}_{2^\ell})^n$ be the circuit being computed over $\mathbb{Z}_{2^\ell}$ such that there are m input wires, n output wires, and each wire can be expressed with ℓ bits. Moreover, assume that the circuit has b binary-multiplication gates. The size of a view in bits is thus given by: $|\mathsf{View}_i| = \ell(m + n + b)$.

ZKBoo Proof Size. Using the above notation, we can now calculate the size of ZKBoo proofs. Let κ be the (classical) security-parameter. The random tapes will be of size κ as mentioned above. Furthermore, let c be the size of the commitments c_i (in bits) for a commitment scheme secure at the given security level. In ZKBoo, hash-based commitments were used and instantiated with SHA-256, and thus $c = 256$. In ZKBoo, the openings D of the commitments contain the value being committed to as well as the randomness used for the commitments. Let s denote the size of the randomness in bits used for each commitment. The size of the output share y_i is the same as the output size of the circuit, $(\ell \cdot n)$. Let t denote the number of parallel repetitions that we must run, and from ZKBoo we know that to achieve soundness error of $2^{-\kappa}$, we must set $t = \lceil \kappa(\log_2 3 - 1)^{-1} \rceil$. The total proof size is given by

$$\begin{aligned}
|p| &= t \cdot [3 \cdot (|y_i| + |c_i|) + 2 \cdot (|\mathsf{View}_i| + |k_i| + s)] \\
&= t \cdot [3 \cdot (\ell n + c) + 2 \cdot (\ell \cdot (m + n + b) + \kappa + s)] \\
&= t \cdot [3c + 2\kappa + 2s + \ell \cdot (5n + 2m + 2b)] \\
&= \lceil \kappa(\log_2 3 - 1)^{-1} \rceil \cdot [3c + 2\kappa + 2s + \ell \cdot (5n + 2m + 2b)]
\end{aligned}$$

3.2 ZKB++

We now present ZKB++, an improved version of ZKBoo with NIZK proofs that are less than half the size of ZKBoo proofs. Moreover, our benchmarks show that this size reduction comes at no extra computational cost.[12]

We present the ZKB++ optimizations in an incremental way over the original ZKBoo protocol.

[12] Our analysis of the original ZKBoo source code uncovered some errors which were corrected in the new implementation.

O1: The Share Function. We make the Share function sample the shares pseudorandomly as:

$$\begin{aligned}(x_1, x_2, x_3) \leftarrow \texttt{Share}(x, k_1, k_2, k_3) :=\\ x_1 &= R_1(0 \cdots |x-1|)\\ x_2 &= R_2(0 \cdots |x-1|)\\ x_3 &= x - x_1 - x_2\end{aligned}$$

where R_i is a pseudorandom generator seeded with k_i, and by $R_i(0 \cdots |x-1|)$ we denote the first $|x|$ bits output by R_i.

We note that sampling in this manner preserves the 2-privacy of the decomposition. In particular, given only two of $\{(k_1, x_1), (k_2, x_2), (k_3, x_3)\}$, x remains uniformly distributed over the choice of the third unopened (k_i, x_i).

We specify the Share function in this manner as it will lead to more compact proofs. For each round, the prover is required to "open" two views. In order to verify the proof, the verifier must be given both the random tape and the input share for each opened view. If these values are generated independently of one another, then the prover will have to explicitly include both of them in the proof. However, with our sampling method, in View_1 and View_2, the prover only needs to include k_i, as x_i can be deterministically computed by the verifier.

The exact savings depend on which views the prover must open, and thus depend on the challenge. The expected reduction in proof size resulting from using the ZKB++ sampling technique instead of the technique used in ZKBoo is $(4t \cdot |x|)/3$ bits.

O2: Not Including Input Shares. Since the input shares are now generated pseudorandomly using the seed k_i, we do not need to include them in the view when $e = 1$. However, if $e = 2$ or $e = 3$, we still need to send one input share for the third view for which the input share cannot be derived from the seed. Since the challenge is generated uniformly at random from $\{1, 2, 3\}$, the expected number of input shares that we'll need to include for a single iteration is $2/3$.

O3: Not Including Commitments. In ZKBoo proofs, the commitments of all three views are sent to the verifier. This is unnecessary as for the two views that are opened, the verifier can recompute the commitment. Only for the third view that the verifier is not given the commitment needs to be explicitly sent.

We stress that there is no lost security here (in some sense we use e as a "commitment to the commitments") as even when the prover sends the commitments, the verifier must check that the prover has sent the correct commitments by hashing the commitments to recompute the challenge. Here too, the verifier checks that the commitments that it computed are the same ones that were used by the prover by hashing them as part of the input to recompute the challenge.

There is also no extra computational cost in this approach – whereas the verifier now must recompute the commitments, in the original ZKBoo protocol, the verifier needed to verify the commitments in step 2 (see Scheme 3 in Appendix A). For the hash-based commitment scheme used in ZKBoo, the function to verify the commitment first recomputes the commitment and thus there is no extra computation.

O4: No Additional Randomness for Commitments. Since the first input to the commitment is the seed value k_i for the random tape, the protocol input to the commitment doubles as a randomization value, ensuring that commitments are hiding. Further, each view included in the commitment must be well randomized for the security of the MPC protocol. In the random oracle model the resulting commitments are hiding (the RO model is needed here since k_i is used both as seed for the PRG and as randomness for the commitment. Since one already needs the RO model to make the proofs non-interactive, there is no extra assumption here).

O5: Not Including the Output Shares. In ZKBoo proofs, as part of a, the output shares y_i are included in the proof. Moreover, for the two views that are opened, those output shares are included a second time.

First, we do not need to send two of the output shares twice. However, we actually do not need to send any output shares at all as they can be deterministically computed from the rest of the proof as follows:

For the two views that are given as part of the proof, the output share can be recomputed from the remaining parts of the view. Essentially, the output share is just the value on the output wires. Given the random tapes and the communicated bits from the binary multiplication gates, all wires for both views can be recomputed.

For the third view, recall that the `Reconstruct` function simply XORs the three output shares to obtain y. But the verifier is given y, and can thus instead recompute the third output share. In particular, given y_i, y_{i+1} and y, the verifier can compute: $y_{i+2} = y + y_i + y_{i+1}$.

Computational Trade-Off. While we would expect some computational cost from recomputing rather than sending the output shares, our benchmarks show that there is no additional computational cost incurred by this modification, perhaps because it is a small part of the overall verification. For the challenge view, View_e, the verifier anyway needs to recompute all of the wire values in order to do the verification, so there is no added cost.

For the second view, View_{e+1}, the verifier must recompute the wire values as well since the verifier will need to compute the values which must be stored as output of the $(2,3)$-decomposition, so there is effectively no cost.

For the third view, the extra cost of recomputing the output share is just two additions in the ring, which is exactly the cost of a single call to `Reconstruct`.

However, in step 2 of the verification in ZKBoo, the verifier has to call `Reconstruct` in order to verify that the three output shares given are correct (see Scheme 3 in Appendix A). But in our optimization, the verifier no longer needs to perform this check as the derivation of the third share guarantees that it will reconstruct correctly. Thus, the verifier is adding one `Reconstruct` but saving one, and thus no cost is incurred.

We note that the outputs will be checked as the y_i's are hashed with H to determine the challenge. The verifier recomputes the challenge and if the y_i values used by the verifier do not match those used by the prover, the challenge will be different (by the collision resistance property of H), and the proof will fail.

O6: Not Including View_e. In step 2 of the proof, the verifier recomputes every wire in View_e and checks as he goes that the received values are correct. However we note that this is not necessary.

The verifier can recompute View_e given just the random tapes k_e, k_{e+1} and the wire values of View_{e+1}. But the verifier does not need to explicitly check that each wire value in View_e is computed correctly. Instead, the verifier will recompute the view, and check the commitments using the recomputed view. By the binding property of the commitment scheme, the commitments will only verify if the verifier has correctly recomputed every value stored in the view.

Notice that this modification reduces the computational time as the verifier does not need to perform part of step 2, i.e., there is no need to check every wire as checking the commitment will check these wires for us. But more crucially, this modification reduces the proof size significantly. There is no need to send the AND wire values for View_e as we can recompute them and check their correctness. Indeed, for this view, the prover only needs to send the input wire value and nothing else.

Putting it All Together: ZKB++ This series of optimizations results in our new protocol ZKB++ which is presented in Scheme 1.

Notice that in ZKB++, the prover explicitly sends the challenge e to the verifier. In the original ZKBoo protocol, the verifier is explicitly given all of the inputs to the challenge random oracle, so it can compute the challenge right away, and then check the proofs. However, in our protocol, the verifier is no longer explicitly given these inputs. Thus our verifier must first recompute all implicitly given values. To be able to compute those values, the challenge e is required which is why we explicitly include e in the proof.

There are 3 possible challenges for each iteration, so the cost of sending e for a t iteration proof is $t \cdot \log_2(3)$.

ZKB++ Proof Size. The expected proof size is

$$\begin{aligned} |p| &= t[|c_i| + 2|k_i| + {}^2\!/\!{}_3|x_i| + b|w_i| + |e_i|] \\ &= t[c + 2\kappa + {}^2\!/\!{}_3\ell m + b\ell + \log_2(3)] \\ &= t[c + 2\kappa + \log_2(3) + \ell \cdot ({}^2\!/\!{}_3 \cdot m + b)] \\ &= \lceil \kappa(\log_2 3 - 1)^{-1} \rceil [c + 2\kappa + \log_2(3) + \ell \cdot ({}^2\!/\!{}_3 \cdot m + b)] \end{aligned}$$

The ZKB++ improvements reduce the proof size compared to ZKBoo by a factor of 2; independent of the concrete circuit.

As an example, we can consider the concrete case of proving knowledge of a SHA-256 pre-image. For this example, we set $\ell = 1$ (for Boolean circuits), $c = 256$ (we use SHA-256 as a commitment scheme), and $s = \kappa$ (the randomness for the commitment in ZKBoo that we eliminated in ZKB++). For the circuit, we use the SHA-256 boolean circuit from [23], for which $m = 512$, $n = 256$, and

For public ϕ and $y \in L_\phi$, the prover has x such that $y = \phi(x)$. The prover and verifier use the hash functions $G(\cdot)$, $H(\cdot)$, and $H'(\cdot)$ modeled as random oracles (H' will be used to commit to the views). The integer t is the number of parallel iterations.

$p \leftarrow \mathtt{Prove}(x)$:

1. For each iteration $r_i, i \in [1,t]$: Sample random tapes $k_1^{(i)}, k_2^{(i)}, k_3^{(i)}$ and simulate the MPC protocol to get an output view $\mathsf{View}_j^{(i)}$ and output share $y_j^{(i)}$. For each player P_j compute
$$(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}) \leftarrow \mathtt{Share}(x, k_1^{(i)}, k_2^{(i)}, k_3^{(i)}) = (G(k_1^{(i)}), G(k_2^{(i)}), x \oplus G(k_1^{(i)}) \oplus G(k_2^{(i)})),$$
$$\mathsf{View}_j^{(i)} \leftarrow \mathtt{Update}(\mathtt{Update}(\cdots \mathtt{Update}(x_j^{(i)}, x_{j+1}^{(i)}, k_j^{(i)}, k_{j+1}^{(i)}) \ldots) \ldots) \ldots),$$
$$y_j^{(i)} \leftarrow \mathtt{Output}(\mathsf{View}_j^{(i)}).$$
Commit $[C_j^{(i)}, D_j^{(i)}] \leftarrow [H'(k_j^{(i)}, x_j^{(i)}, \mathsf{View}_j^{(i)}), k_j^{(i)} || \mathsf{View}_j^{(i)}]$, and let $a^{(i)} = (y_1^{(i)}, y_2^{(i)}, y_3^{(i)}, C_1^{(i)}, C_2^{(i)}, C_3^{(i)})$.
2. Compute the challenge: $e \leftarrow H(a^{(1)}, \ldots, a^{(t)})$. Interpret the challenge such that for $i \in [1,t]$, $e^{(i)} \in \{1,2,3\}$
3. For each iteration r_i, $i \in [1,t]$: let $b^{(i)} = (y_{e^{(i)}+2}^{(i)}, C_{e^{(i)}+2}^{(i)})$ and set
$$z^{(i)} \leftarrow \begin{cases} (\mathsf{View}_2^{(i)}, k_1^{(i)}, k_2^{(i)}) & \text{if } e^{(i)} = 1, \\ (\mathsf{View}_3^{(i)}, k_2^{(i)}, k_3^{(i)}, x_3^{(i)}) & \text{if } e^{(i)} = 2, \\ (\mathsf{View}_1^{(i)}, k_3^{(i)}, k_1^{(i)}, x_3^{(i)}) & \text{if } e^{(i)} = 3. \end{cases}$$
4. Output $p \leftarrow [e, (b^{(1)}, z^{(1)}), (b^{(2)}, z^{(2)}), \cdots, (b^{(t)}, z^{(t)})]$.

$b \leftarrow \mathtt{Verify}(y, p)$:

1. For each iteration $r_i, i \in [1,t]$: Run the MPC protocol to reconstruct the views, input and output shares that were not explicitly given as part of the proof p. In particular:
$$x_{e^{(i)}}^{(i)} \leftarrow \begin{cases} G(k_1^{(i)}) & \text{if } e^{(i)} = 1, \\ G(k_2^{(i)}) & \text{if } e^{(i)} = 2, \\ x_3^{(i)} \text{ from } z^{(i)} & \text{if } e^{(i)} = 3. \end{cases} \quad x_{e^{(i)}+1}^{(i)} \leftarrow \begin{cases} G(k_2^{(i)}) & \text{if } e^{(i)} = 1, \\ x_3^{(i)} \text{ from } z^{(i)} & \text{if } e^{(i)} = 2, \\ G(k_1^{(i)}) & \text{if } e^{(i)} = 3. \end{cases}$$
Obtain $\mathsf{View}_{e^{(i)}+1}^{(i)}$ from $z^{(i)}$ and compute
$$\mathsf{View}_e^{(i)} \leftarrow \mathtt{Update}(\ldots \mathtt{Update}(x_e^{(i)}, x_{e+1}^{(i)}, k_e^{(i)}, k_{e+1}^{(i)}) \ldots),$$
$$y_j^{(i)} \leftarrow \mathtt{Output}(\mathsf{View}_j^{(i)}) \text{ for } j \in \{e^{(i)}, e^{(i)}+1\},\ y_{e^{(i)}+2}^{(i)} \leftarrow y \oplus y_{e^{(i)}}^{(i)} \oplus y_{e^{(i)}+1}^{(i)}$$
Compute the commitments for views $\mathsf{View}_{e^{(i)}}^{(i)}$ and $\mathsf{View}_{e^{(i)}}^{(i)}$. For $j \in \{e^{(i)}, e^{(i)}+1\}$:
$$[C_j^{(i)}, D_j^{(i)}] \leftarrow [H'(k_j^{(i)}, x_j^{(i)}, \mathsf{View}_j^{(i)}), k_j^{(i)} || \mathsf{View}_j^{(i)}]$$
Let $a'^{(i)} = (y_1^{(i)}, y_2^{(i)}, y_3^{(i)}, C_1^{(i)}, C_2^{(i)}, C_3^{(i)})$ and note that $y_{e^{(i)}+2}^{(i)}$ and $C_{e^{(i)}+2}^{(i)}$ is a part of $z^{(i)}$.
2. Compute the challenge: $e' \leftarrow H(a'^{(1)}, \ldots, a'^{(t)})$. If, $e' = e$, output Accept, otherwise output Reject.

Scheme 1: The **ZKB++** proof system, made non-interactive using the Fiat-Shamir transform.

$b = 22272$. Given these parameters, if we set $\kappa = 128$, then the ZKB++ proof size is 618 kilobytes, which is only 48% of ZKBoo proof size (1287 kilobytes). At the 80-bit security level, the ZKB++ proof size is 385 kilobytes, and at the 40-bit security level, the proof size is 193 kilobytes. For all these figures, we used 256-bit commitments, and thus in practice they may be slightly reduced by using a weaker commitment scheme.

ZKB++ Security. From our argumentation above we conclude that the security of ZKBoo directly implies security of ZKB++ in the (Q)ROM.

4 The Fish Signature Scheme

The FS transform is an elegant way to obtain EUF-CMA secure signature schemes. The basic idea is similar to constructing NIZK proofs from Σ-protocols, but the challenge c is generated by hashing the prover's first message r and the message m to be signed, i.e., $c \leftarrow H(r, m)$. In the following we will index the non-interactive PPT algorithms $(\texttt{Prove}_H, \texttt{Verify}_H)$ by the hash function H, which we model as a random oracle. Let us consider a language L_R with associated witness relation R of pre-images of a one-way function $f_k : \mathsf{D}_\kappa \rightarrow \mathsf{R}_\kappa$, sampled uniformly at random from a family of one-way functions $\{f_k\}_{k \in \mathsf{K}_\kappa}$, indexed by key k and security parameter κ:

$$((y, k), x) \in R \iff y = f_k(x).$$

Henceforth, we may use $\{f_k\}$ for brevity. The function family $\{f_k\}$ could be any one-way function family, but since we found that function families based on block ciphers gave the most efficient signatures, we tailor our description to this choice of $\{f_k\}$. Here we have that

$$f_k(x) := \mathsf{Enc}(x, k),$$

where $\mathsf{Enc}(x, k)$ denotes the encryption of a single block $k \in \{0, 1\}^{c \cdot \kappa}$ with respect to key $x \in \{0, 1\}^{c \cdot \kappa}$. One can sample a one-way function $\{f_k\}$ with respect to security parameter κ uniformly at random by sampling a uniformly random block $k \in \{0, 1\}^{c \cdot \kappa}$. In Appendix D we formally argue that we can use a block cipher (viewed as a PRF) in this way to instantiate an OWF. In the classical setting we set $c = 1$, whereas we set $c = 2$ in the post-quantum setting to account for the generic speedup imposed by Grover's algorithm [52]. The rationale for using a random instead of a fixed block k when creating the signature keypair is to improve security against multi-user key recovery attacks and generic time-memory trade-off attacks like [54]. To reduce the size of the public key, one could choose a smaller value that is unique per user, or use a fixed value (with a potential decrease in security). Since public keys in our schemes are small (at most 64 bytes), our design uses a full random block.

When using ZKBoo to prove knowledge of such a pre-image, we know [44] that this Σ-protocol provides 3-special soundness. We apply the FS transform to this Σ-protocol to obtain an EUF-CMA secure signature scheme. In the so-obtained signature scheme the public verification key pk contains the image y and the value k determining f_k. The secret signing key sk is a random value x from D_κ. The corresponding signature scheme, dubbed Fish, is illustrated in Scheme 2.

$\underline{\mathsf{Gen}(1^\kappa)}$: Choose $k \xleftarrow{R} \mathsf{K}_\kappa$, $x \xleftarrow{R} \mathsf{D}_\kappa$, compute $y \leftarrow f_k(x)$, set $\mathsf{pk} \leftarrow (y, k)$ and $\mathsf{sk} \leftarrow (\mathsf{pk}, x)$ and return $(\mathsf{sk}, \mathsf{pk})$.

$\underline{\mathsf{Sign}(\mathsf{sk}, m)}$: Parse sk as (pk, x), compute $p = (r, s) \leftarrow \mathtt{Prove}_H((y, k), x)$ and return $\sigma \leftarrow p$, where internally the challenge is computed as $c \leftarrow H(r, m)$.

$\underline{\mathsf{Verify}(\mathsf{pk}, m, \sigma)}$: Parse pk as (y, k), and σ as $p = (r, s)$. Return 1 if the following holds, and 0 otherwise:

$$\mathtt{Verify}_H((y, k), p) = 1,$$

where internally the challenge is computed as $c \leftarrow H(r, m)$.

Scheme 2: Generic description the Fish and Picnic signature schemes. In both schemes Prove is implemented with ZKB++, in Fish it is made non-interactive with the FS transform, while in Picnic, Unruh's transform is used.

If we view ZKBoo as a canonical identification scheme that is secure against passive adversaries one just needs to keep in mind that most definitions are tailored to 2-special soundness, and the 3-special soundness of ZKBoo requires an additional rewind. In particular, an adapted version of the proof of [61, Theorem 8.2] which considers this additional rewind attests the security of Scheme 2. The security reduction, however, is a non-tight one, like most signature schemes constructed from Σ-protocols.[13] We obtain the following:

Corollary 1. *Scheme 2 instantiated with* ZKB++ *and a secure one-way function yields an* EUF-CMA *secure signature scheme in the ROM.*

5 The Picnic Signature Scheme

The Picnic signature scheme is the same as Fish, except for the transform used to make ZKB++ noninteractive. Unruh [83] presents an alternative to the FS transform that is provably secure in the QROM. Indeed, Unruh even explicitly presents a construction for a signature scheme and proves its security given a secure a $\Sigma-$protocol. Unruh's construction requires a $\Sigma-$protocol and a hard instance generator, but he does not give an instantiation. We use his approach to argue that with a few modifications, our signature scheme is also provably secure in the QROM. One interesting aspect is that, while on first observation Unruh's transform seems much more expensive than the standard FS transform, we show how to make use of the structure of ZKB++ to reduce the cost significantly.

[13] There are numerous works on signatures from (three move) identification schemes [73,77,1,2,62,12,32]. Unfortunately existing proof techniques do not give tight security reductions.

Unruh's Transform: Overview. At a high level, Unruh's transform works as follows: Given a Σ-protocol with challenge space C, an integer t, a statement x, and a random permutation G, the prover will

1. Run the first phase of the Σ-protocol t times to produce $r_1, \ldots, r_t$.
2. For each $i \in \{1, \ldots, t\}$, and for each $j \in C$, compute the response s_{ij} for r_i and challenge j. Compute $g_{ij} = G(s_{ij})$.
3. Compute $H(x, r_1, \ldots, r_t, g_{11}, \ldots, g_{t|C|})$ to obtain a set of indices $J_1, \ldots, J_t$.
4. Output $\pi = (r_1, \ldots, r_t, s_{1J_1}, \ldots, s_{tJ_t}, g_{11}, \ldots, g_{t|C|})$.

Similarly, the verifier will verify the hash, verify that the given s_{iJ_i} values match the corresponding g_{iJ_i} values, and that the s_{iJ_i} values are valid responses w.r.t. the r_i values.

Informally speaking, in Unruh's security analysis, zero knowledge follows from HVZK of the underlying Σ-protocol: the simulator just generates t transcripts and then programs the random oracle to get the appropriate challenges. The proof of knowledge property is more complex, but the argument is that any adversary who has non-trivial probability of producing an accepting proof will also have to output some g_{ij} for $j \neq J_i$ which is a correct response for a different challenge - then the extractor can invert G and get the second response, which by special soundness allows it to produce a witness.

To instantiate the function G in the protocol, Unruh shows that one does not need a random oracle that is actually a permutation. Instead, as long as the domain and co-domain of G have the same length, it can be used, since it is indistinguishable from a random permutation.

Applying the Unruh transform to ZKB++: The Direct Approach. We can apply Unruh to ZKB++ in a relatively straightforward manner by modifying our protocol. Although ZKB++ has 3-special soundness, whereas Unruh's transform is only proven for Σ-protocols with 2-special soundness, the proof is easily modified to 3-special soundness.

Since ZKB++ has 3-special soundness, we would need at least three responses for each iteration. Moreover, since there only are three possible challenges in ZKB++, we would run Unruh's transform with $C = \{1, 2, 3\}$, i.e., every possible challenge and response. We would then proceed as follows:

Let $G : \{0,1\}^{|s_{ij}|} \to \{0,1\}^{|s_{ij}|}$ be a hash function modeled as a random oracle.[14] Non-interactive ZKB++ proofs would then proceed as follows:

1. Run the first ZKB++ phase t times to produce $r_1, \ldots, r_t$.
2. For each $i \in \{1, \ldots, t\}$, and for each $j \in 1, 2, 3$, compute the response s_{ij} for r_i and challenge j. Compute $g_{ij} = G(s_{ij})$.
3. Compute $H(x, r_1, \ldots, r_t, g_{11}, \ldots, g_{t3})$ to obtain a set of indices $J_1, \ldots, J_t$.
4. Output $\pi = (r_1, \ldots, r_t, s_{1J_1}, \ldots, s_{tJ_t}, g_{11}, \ldots, g_{t3})$.

[14] Actually, the size of the response changes depending on what the challenge is. If the challenge is 0, the response is slightly smaller as it does not need to include the extra input share. So more precisely, this is actually two hash functions, G_0 used for the 0-challenge response and $G_{1,2}$ used for the other two.

While this works, it comes as a significant overhead in the size of the proof. That is, we have to additionally include $g_{11}, \ldots, g_{t3}$. Each g_{ij} is a permutation of an output share and there are $3t$ such values, so in particular the extra overhead would yield a proof size of

$$\begin{aligned}
&t \cdot [c + 2\kappa + \log_2(3) + \ell \cdot ({}^2\!/\!_3 \cdot m + b)] + \\
&3t \cdot [2\kappa + \ell \cdot ({}^2\!/\!_3 \cdot m + b)] = \\
&t \cdot [c + 8\kappa + log_2(3) + \ell \cdot ({}^8\!/\!_3 m + 4b)].
\end{aligned}$$

Since for most functions, the size of the proof is dominated by $t \cdot \ell b$, this proof is roughly four times as large as in the FS version. To this end, we again introduce some optimizations.

O1: Making Use of Overlapping Responses. We can make use of the structure of the ZKB++ proofs to achieve a significant reduction in the proof size. Although we refer to three separate challenges, in the case of the ZKB++ protocol, there is a large overlap between the contents of the responses corresponding to these challenges. In particular, there are only three distinct views in the ZKB++ protocol, two of which are opened for a given challenge.

Instead of computing a permutation of each *response*, s_{ij}, we can compute a permutation of each *view*, v_{ij}. For each $i \in \{1, \ldots, t\}$, and for each $j \in \{1, 2, 3\}$, the prover computes $g_{ij} = G(v_{ij})$.

The verifier checks the permuted value for each of the two views in the response. In particular, for challenge $i \in \{1, 2, 3\}$, the verifier will need to check that $g_{ij} = G(v_{ij})$ and $g_{i(j+1)} = G(v_{i(j+1)})$.

O2: Omit Re-Computable Values. Moreover, since G is a public function, we do not need to include $G(v_{ij})$ in the transcript if we have included v_{ij} in the response. Thus for the two views (corresponding to a single challenge) that the prover sends as part of the proof, we do not need to include the permutations of those views. We only need to include $G(v_{i(j+2)})$, where $v_{i(j+2)}$ is the view that the prover does not open for the given challenge.

Putting it Together: New Proof Size. Combining these two modifications yields a major reduction in proof size. For each of the t iterations of ZKB++, we include just a single extra $G(v)$ than we would in the FS transform.

As G is a permutation, the per-iteration overhead of ZKB++/Unruh over ZKB++/FS is the size of a single view. This overhead is less that one-third of the overhead that would be incurred from the naive application of Unruh as described before. In particular, the expected proof size of our optimized version is then

$$\begin{aligned}
&t \cdot [c + 2\kappa + \log_2(3) + \ell \cdot ({}^2\!/\!_3 \cdot m + b)] + \\
&t \cdot [\kappa + \ell \cdot ({}^1\!/\!_3 \cdot m + b)] = \\
&t \cdot [c + 3\kappa + log_2(3) + \ell \cdot (m + 2b)].
\end{aligned}$$

The overhead depends on the circuit. For LowMC, we found the overhead ranges from 1.6 to 2 compared to the equivalent ZKB++/FS proof.

Security of the Modified Unruh Transform. For zero knowledge, we can take the same approach as in Unruh [84]: to simulate the proof we choose the set of challenges $J_1, \ldots, J_t$, run the (2,3)-decomposition simulator to obtain views for each pair of dishonest parties J_i, J_{i+1}, honestly generate g_{iJ_i} and $g_{iJ_{i+1}}$ and the commitments to those views, and choose $g_{J_{i+2}}$ and the corresponding commitment at random. Then we program the random oracle to output $J_1, \ldots, J_t$ on the resulting tuple. The analysis follows exactly as in [84].

For the soundness argument, our protocol has two main differences from Unruh's general version: (1) the underlying protocol we use only has 3-special soundness, rather than the normal 2-special soundness, and (2) we have one commitment for each view, and one $G(v)$ for each view, rather than having a separate $G(view_i, view_{i+1})$ for each i.

As mentioned above, the core of Unruh's argument [84, Lemma 17], says that the probability that the adversary can find a proof such that the extractor cannot extract but the proof still verifies is negligible.

For our case, the analysis is as follows: For a given tuple of commitments $r_1 \ldots r_t$, and G-values $g_{11}, g_{t|C|}$ that is queried to the random oracle either one of the following is true: (1) There is some i for which $(G^{-1}(g_{i1}),\ G^{-1}(g_{i2}))$, $(G^{-1}(g_{i2}),\ G^{-1}(g_{i3}))$, $(G^{-1}(g_{i3}),\ G^{-1}(g_{i1}))$, are valid responses for challenges $1, 2, 3$ respectively[15], or (2) For all i at least one of these pairs is not a valid response. In particular this means that if this is the challenge produced by the hash function, $\mathcal{A}$ will not be able to produce an accepting response. From that, we can argue that if the extractor cannot extract from a given tuple, then the probability (over the choice of a RO) that there exists an accepting response for $\mathcal{A}$ to output is at most $(2/3)^t$. Then, we can rely on [84, Lemma 7], which tells us that given q_H queries, the probability that $\mathcal{A}$ produces a tuple from which we cannot extract but $\mathcal{A}$ can produce an accepting response is at most $2(q_H + 1)(2/3)^t$.

The rest of our argument can proceed exactly as in Unruh's proof and we obtain the following:

Corollary 2. *Scheme 2 instantiated with* ZKB++*, a secure permutation and one-way function yields an* EUF-CMA *secure signature scheme in the QROM.*

The full proof is given in Appendix F. The security reduction in our proof is non-tight, the gap is proportional to the number of RO queries.

Unruh's Transform with Constant Overhead? We conjecture that we may be able to further reduce the overhead of Unruh's transform to a fixed size that does not depend on the circuit being used. We leave this as a conjecture for now as it does not follow from Unruh's proof, and we have not proved it.

If we were to include just the hash using G of the seeds (and the third input share that is not derivable from its seed), it seems that this would be enough for the extractor to produce a witness. Combining this with the previous optimizations, we only need to explicitly give the extractor a permutation of the

[15] In fact G is not exactly a permutation, but we ignore that here. We can make this formal exactly as in Unruh's proof, by considering the set of pre-images.

input share of the third view. For the first two views, the views are communicated in the open, and the extractor can compute the permutation himself. This would reduce the overhead when compared to FS from about 1.6x to 1.16x.

6 Selecting an Underlying Primitive

We require one or more symmetric primitives suitable to instantiate a one-way function. We now first investigate how choosing a primitive with certain properties impacts the instantiations of our schemes. From this, we derive concrete requirements, and present our choice, LowMC.

6.1 Survey of Suitable Primitives

The signature size depends on constants that are close to the security expectation (cf. Section 7 for our choices). The only exceptions are the number of binary multiplication gates, and the size of the rings, which all depend on the choice of the primitive. Hence we survey existing designs that can serve as a one-way function subsequently.

Standardized General-Purpose Primitives. The smallest known Boolean circuit of AES-128 needs 5440 AND gates, AES-192 needs 6528 AND gates, and AES-256 needs 7616 AND gates [20]. An AES circuit in $\mathbb{F}_{2^4}$ might be more efficient in our setting, as in this case the number of multiplications is lower than 1000 [25]. This results in an impact on the signature size that is equivalent to 4000 AND gates. Even though collision resistance is often not required, hash functions like SHA-256 are a popular choice for proof-of-concept implementations. The number of AND gates of a single call to the SHA-256 compression function is about 25000 and a single call to the permutation underlying SHA-3 is 38400.

Lightweight Ciphers. Most early designs in this domain focused on small area when implemented in hardware where an XOR gate is by a small factor larger than an AND or NAND gate. Notable designs with a low number of AND gates at the 128-bit security level are the block ciphers Noekeon [29] (2048) and Fantomas [50] (2112). Furthermore, one should mention Prince [19] (1920), or the stream cipher Trivium [33] (1536 AND gates to compute 128 output bits) with 80-bit security.

Custom Ciphers with a Low Number of Multiplications. Motivated by applications in SHE/FHE schemes, MPC protocols and SNARKs, recently a trend to design symmetric encryption primitives with a low number of multiplications or a low multiplicative depth started to evolve. This is a trend we can take advantage of.

We start with the LowMC [6] block cipher family. In the most recent version of the design [4], the number of AND gates can be below 500 for 80-bit security, below 800 for 128-bit security, and below 1400 for 256-bit security. The stream cipher Kreyvium [24] needs similarly to Trivium 1536 AND gates to compute 128 output bits, but offers a higher security level of 128 bits. Even though FLIP [69] was designed to have especially low depth, it needs hundreds of AND gates per bit and is hence not competitive in our setting.

Last but not least there are the block ciphers and hash functions around MiMC [5] which need less than $2 \cdot s$ multiplications for s-bit security in a field of size close to 2^s. Note that MiMC is the only design in this category which aims at minimizing multiplications in a field larger than $\mathbb{F}_2$. However, since the size of the signature depends on both the number of multiplications and the size of the field, this leads to a factor $2s^2$ which, for all arguably secure instantiations of MiMC, is already larger than the number of AND gates in the AES circuit.

LowMC has two important advantages over other designs: It has the lowest number of AND gates for every security level: The closest competitor Kreyvium needs about twice as many AND gates and only exists for the 128-bit security level. The fact that it allows for an easy parameterization of the security level is another advantage. We hence use LowMC for our concrete proposal and discuss it in more detail in the following.

6.2 LowMC

LowMC is a flexible block cipher family based on a substitution-permutation network. The block size n, the key size k, the number of 3-bit S-boxes m in the substitution layer and the allowed data complexity d of attacks can independently be chosen. 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. The number of rounds r needed to achieve the goals is then determined as a function of all these parameters. For the sake of completeness we include a brief description of LowMC in Appendix B.

To minimize the number of AND gates for a given k and d, we want to minimize $r \cdot m$. A natural strategy would be to set m to 1, and to look for an n that minimizes r. Examples of such an approach are already given in the document describing version 2 of the design [4]. In our setting, this approach may not lead to the best results, as it ignores the impact of the large amount of XOR operations it requires. To find the most suitable parameters, we thus explore a larger range of values for m.

Whenever we want to instantiate our signature scheme with LowMC with s-bit PQ-security, we set $k = n = 2 \cdot s$. This choice to double the parameter in the quantum setting takes into account current knowledge of quantum-cryptanalysis for models that are very generous to the attacker [60,59]. Note that setting $s = 64, 96, 128$ matches the requirements of the upcoming NIST selection process[16] for security levels 1, 3 and 5, respectively. Section 7 gives benchmarks for levels 1, 3, and 5.

Furthermore, we observe that the adversary only ever sees a single plaintext-ciphertext pair. In the security proof given in Appendix D, we build a distin-

[16] `http://csrc.nist.gov/groups/ST/post-quantum-crypto/`

guisher that only needs to see one additional pair. This is why we can set the data complexity $d = 1$.[17]

7 Implementation and Parameters

We pursue two different directions. First, we present a general purpose implementation for the Fish signature scheme.[18] This library exposes an API to generate LowMC instances for a given parameter set, as well as an easy to use interface for key generation, signature generation/verification in both schemes. Using this library we explore the whole design space of LowMC to find the most suitable instances. Second, we present a library which implements the Picnic signature scheme[19]. This implementation is parameterized with the previously selected LowMC instance, since the QROM instantiation imposes a constant overhead which is independent of the LowMC instance. Both libraries are implemented in C using the OpenSSL[20] and m4ri[21] libraries. We have released both our libraries as open source under the MIT License.

7.1 Implementation of Building Blocks

The building blocks in the protocol are instantiated similar to the implementation of ZKBoo [44]. In Appendix C and D, we give more formal arguments regarding our choices.

PRG. Random tapes are generated pseudorandomly using AES in counter mode, where the keys are generated using OpenSSL's secure random number generator. In the linear decomposition of the AND gates we use a function that picks the random bits from the bit stream generated using AES. Since the number of AND gates is known a-priori, we can pre-compute all random bits at the beginning. Concretely, we assume that AES-256 in counter mode provides 128 bits of PRG security, when used to expand 256-bit seeds to outputs ≈ 1kB in length.

Commitments. The commitment function (used to commit to the views) is implemented using SHA-256.

Challenge Generation. For both schemes the challenge is computed with a hash function $H : \{0,1\}^* \rightarrow \{0,1,2\}^t$ implemented using SHA-256 and rejection sampling: we split the output bits of SHA-256 in pairs of two bits and reject all pairs with both bits set.

[17] d is given in units of $\log_2(n)$, where n is the number of pairs. Thus setting $d = 1$ corresponds to 2-pairs, which is exactly what we need for our signature schemes.

[18] https://github.com/IAIK/fish-begol

[19] https://github.com/Microsoft/Picnic

[20] https://openssl.org

[21] https://bitbucket.org/malb/m4ri

One-Way Function. The OWF function family $\{f_k\}_{k \in \mathsf{K}_\kappa}$ used for key generation in both signature schemes is instantiated with LowMC. Concretely, we instantiate $\{f_k\}$ using a block cipher with

$$f_k(x) := \mathsf{Enc}(x, k),$$

where $\mathsf{Enc}(x, k)$ denotes the LowMC encryption of a single block $k \in \{0,1\}^\kappa$ with respect to key $x \in \{0,1\}^\kappa$. For such an instantiation we assume that we have $\kappa/2$ bit security. In Appendix D we provide further details on this choice. There, to make our results more general, we also show that a block cipher with $k = n = 2s$ when viewed as a PRF can be used as an OWF with $2s$-bit classical security, and thus gives us the s-bit post-quantum security that we desire. Our implementations support multiple LowMC parameter sets.

Function G. As explained in Section 5, G may be implemented with a random function with the same domain and range. We implement $G(x)$ as $h(0\|x)\|h(1\|x)\ldots$, where h is SHA-256 and the output length is $|x|$.

Hash Function Security. We make the following concrete assumptions for the security of our schemes. We assume that SHA-256 provides 128 bits of pre-image resistance against quantum adversaries. For collision resistance, when considering quantum algorithms, in theory it may be possible to find collisions using a generic algorithm of Brassard et al. [21] with cost $O(2^{n/3})$. A detailed analysis of the costs of the algorithm in [21] by Bernstein [16] found that in practice the quantum algorithm is unlikely to outperform the $O(2^{n/2})$ classical algorithm. Multiple cryptosystems have since made the assumption that standard hash functions with n-bit digests provide $n/2$ bits of collision resistance against quantum attacks (for examples, see papers citing [16]). We make this assumption as well, and in particular, that SHA-256 provides 128 bits of PQ collision-resistance.

7.2 Circuit for LowMC

For the linear (2,3)-decomposition we view LowMC as circuit over $\mathbb{F}_2$. The circuit consists only of AND and XOR gates. The number of bits we have to store per view is $3 \cdot r \cdot m$, where r is the number of rounds and m is the number of S-boxes.

Since the affine layer of LowMC only consists of AND and XOR operations, it benefits from using block sizes such that all computations of this layer can be performed using SIMD instruction sets like SSE2, AVX2 and NEON, i.e., 128-bit or 256-bit. Since our implementation uses (arrays of) native words to store the bit vectors, the implementation benefits from a choice of parameters such that $3 \cdot m$ is close to the word size. This choice allows us to maximize the number of parallel S-box evaluations in the bitsliced implementation.

7.3 Experimental Setup and Results

Our experiments were performed on an Intel Core i7-4790 CPU (4 cores with 3.60 GHz) and 16 GB RAM running Ubuntu 16.10. Henceforth, we target the 128 bit post-quantum setting.

Number of Parallel Repetitions. While we already established that ZKB++ is a suitable Σ-protocol (see the discussion at the end of Section 3.2), we must set the number of parallel repetitions to achieve the desired soundness error. For a single repetition we have a soundness error of $2/3$, which means that we need 219 parallel repetitions for 128-bit security ($(3/2)^{219} \geq 2^{128}$). For 128-bit PQ security, we must set our repetition count to $t := 438$. This is double the repetition count required for classical security due to Grover's algorithm [52]. To see the effects of the search algorithm, an adversary at first computes t views such that it can answer two of the three possible challenges honestly for each view. Considering the possible permutations of the individual views, the adversary is thus able to answer 2^t out of the 3^t challenges. Grover's algorithm is then tasked to find a permutation of the views such that they correspond to one of the 2^t challenges. Out of the 2^t permutations, the expected number of solutions is $(4/3)^t$, hence Grover's algorithm reduces the time to find a solution to $(3/2)^{t/2}$. So for the 128-bit PQ security level, we require t be large enough to satisfy $(3/2)^{t/2} \geq 2^{128}$, and so $t = 438$ is the smallest possible repetition count.

Each of the parallel repetitions are largely independent. Thus, we can split the signature generation/verification among multiple cores. In Appendix E we discuss the benefits of using multiple cores.

Selection of the Most Suitable LowMC Instances. We now explore the design space of LowMC. Figure 1 shows that choosing a concrete LowMC instance allows a trade-off between computational efficiency and signature size, parameterized by the number of rounds and by the number of S-boxes.

Using the notation [blocksize]-[keysize]-[#sboxes]-[#rounds], we recommend the 256-256-10-38 instance as a good balance between speed and size.

To support our choice of LowMC, we note that running the implementation for the SHA-256 circuit from [44] with $t = 438$ repetitions on the same machine yields roughly 2.7MB proof size, signing times of 237ms, and verification times of 137ms. Informally speaking, this can be seen as a baseline instantiation of our scheme Fish with SHA-256 instead of LowMC and ZKBoo instead of ZKB++ (cf. Table 1 for our results when using LowMC).

7.4 Comparison with Related Work

To compare our schemes to other post-quantum signature candidates, we focused on those that have a reference implementation available and ran the benchmarks on our machine. Table 1 gives an overview of the results, including MQDSS [56], the lattice based schemes TESLA [7][22], ring-TESLA [3] and BLISS [36], the hash-based scheme SPHINCS-256 [17], the supersingular isogeny-based scheme SIDHp751 [87], and also give sizes for the code-based scheme FS-Véron [86]

[22] Due to an erroneous security analysis the scheme has been revised [8]. But since this happened after we performed our benchmark computations, we present the performance of the original TESLA scheme.

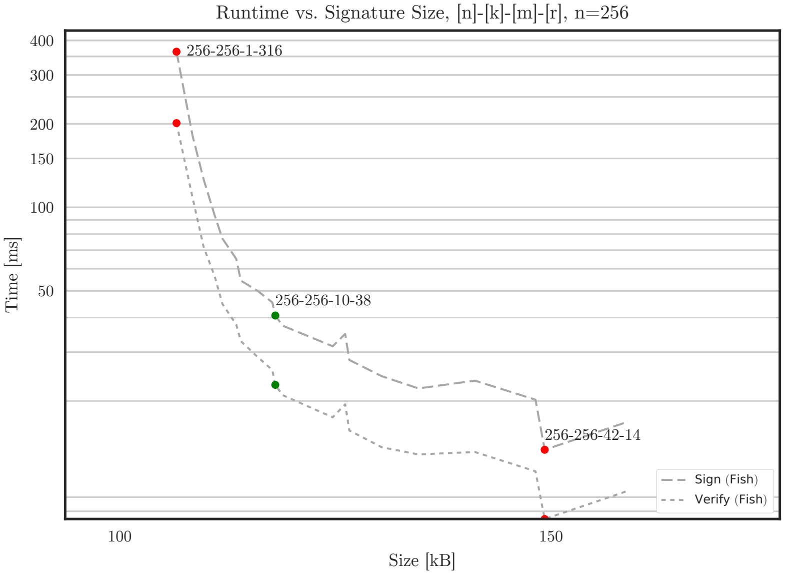


Fig. 1. Measurements for instance selection (128-bit post-quantum security, average over 100 runs).

to complete the picture.[23] For our schemes, we include LowMC instances with 128, 192, and 256 bit block- and keysize for levels 1, 3, and 5, respectively. For all three levels we use 10 S-boxes for LowMC. Additionally, for level 5 we also include the extreme points from the instance selection. Note however, that the implementations for levels 1 and 5 profit more from our SIMD-based optimizations then the implementation for level 3.

Our implementation is a highly parameterizable implementation, flexible enough to cover the entire design spectrum of our approaches. In contrast, the implementations of other candidates used for comparison come with a highly optimized implementation targeting a specific security level (and often also specific instances). Thus, our timings are more conservative than the ones of the other schemes. Yet, while timings and sizes can largely not compete with efficient lattice-based schemes using ideal lattices, they are comparable to all other existing post-quantum candidates. We want to stress that ideal lattices have not been investigated nearly as deeply as standard lattices and thus there is less confidence in the assumptions (cf. [75]) and also the choice of parameters of these schemes can be seen as quite aggressive.

[23] Key sizes and signature sizes from BLISS were taken from [36], as they were not readily available in the implementation. Sizes for FS-Véron are taken from `https://pqcrypto.eu.org/mini.html`.

Scheme	Gen	Sign	Verify	|sk|	|pk|	|σ|	Model
	[ms]	[ms]	[ms]	[bytes]	[bytes]	[bytes]	
Fish-L1-10-20	0.01	3.94	1.69	16	32	37473	ROM
Fish-L3-10-30	0.01	51.33	32.01	24	48	73895	ROM
Fish-L5-1-316	0.01	364.11	201.17	32	64	108013	ROM
Fish-L5-10-38	0.01	29.73	17.46	32	64	118525	ROM
Fish-L5-42-14	0.01	13.27	7.45	32	64	152689	ROM
Picnic-L5-10-38	0.01	31.31	16.30	32	64	195458	QROM
MQ 5pass	0.96	7.21	5.17	32	74	40952	ROM
SPHINCS-256	0.82	13.44	0.58	1088	1056	41000	SM
BLISS-I	44.16	0.12	0.02	2048	7168	5732	ROM
Ring-TESLA*	16k	0.06	0.03	12288	8192	1568	ROM
TESLA-768*	48k	0.65	0.36	3216k	4128k	2336	(Q)ROM
FS-Véron	n/a	n/a	n/a	32	160	129024	ROM
SIDHp751	16.41	7.3k	5.0k	48	768	141312	QROM

Table 1. Timings and sizes of private keys (sk), public keys (pk) and signatures (σ) at the post-quantum 128-bit security level. *An errata to [3] says that this parameter set is not supported by the security analysis (due to a flaw).

8 Summary

We have proposed two post-quantum signature schemes, i.e., Fish and Picnic. On our way, we optimize ZKBoo to obtain ZKB++. For Fish, we then apply the FS transform ZKBoo, whereas we optimize the Unruh transform and apply it to ZKB++ for Picnic. Fish is secure in the ROM, while Picnic is secure in the QROM. ZKB++ optimizes ZKBoo by reducing the proof sizes by a factor of two, at no additional computational cost. While this is of independent interest as it yields more compact (post-quantum) zero-knowledge proofs for any circuit, it also decreases our signature sizes. Our work establishes a new direction to design post-quantum signature schemes and we believe that this is an interesting direction for future work, e.g., by the design of new symmetric primitives especially focusing on optimizing the metrics required by our approach. Also, as ZKBoo/ZKB++ are still relatively young it is likely that we will see further improvements in the next few years (for a recent example see [78]).

Acknowledgments. D. Derler, S. Ramacher, C. Rechberger, and D. Slamanig have been supported by H2020 project Prismacloud, grant agreement n°644962. C. Rechberger has additionally been supported by EU H2020 project PQCRYPTO, grant agreement n°645622. Steven Goldfeder is supported by the NSF Graduate Research Fellowship under grant number DGE 1148900. C. Orlandi has been supported by COST Action IC1306 and the Danish Council for Independent Research.

References

1. Abdalla, M., An, J. H., Bellare, M., and Namprempre, C. From identi-

fication to signatures via the fiat-shamir transform: Minimizing assumptions for security and forward-security. In *EUROCRYPT* (2002).

2. Abdalla, M., Fouque, P., Lyubashevsky, V., and Tibouchi, M. Tightly-secure signatures from lossy identification schemes. In *EUROCRYPT* (2012).
3. Akleylek, S., Bindel, N., Buchmann, J. A., Krämer, J., and Marson, G. A. An efficient lattice-based signature scheme with provably secure instantiation. In *AFRICACRYPT* (2016).
4. Albrecht, M., Rechberger, C., Schneider, T., Tiessen, T., and Zohner, M. Ciphers for MPC and FHE. Cryptology ePrint Archive, Report 2016/687, 2016.
5. Albrecht, M. R., Grassi, L., Rechberger, C., Roy, A., and Tiessen, T. MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In *ASIACRYPT* (2016), pp. 191–219.
6. Albrecht, M. R., Rechberger, C., Schneider, T., Tiessen, T., and Zohner, M. Ciphers for MPC and FHE. In *EUROCRYPT* (2015).
7. Alkim, E., Bindel, N., Buchmann, J., Dagdelen, Ö., and Schwabe, P. Tesla: Tightly-secure efficient signatures from standard lattices. Cryptology ePrint Archive, Report 2015/755, 2015.
8. Alkim, E., Bindel, N., Buchmann, J. A., Dagdelen, Ö., Eaton, E., Gutoski, G., Krämer, J., and Pawlega, F. Revisiting TESLA in the quantum random oracle model. In *PQCrypto 2017* (2017), pp. 143–162.
9. Bai, S., and Galbraith, S. D. An improved compression technique for signatures based on learning with errors. In *CT-RSA* (2014).
10. Bansarkhani, R. E., and Buchmann, J. A. Improvement and efficient implementation of a lattice-based signature scheme. In *SAC* (2013).
11. Barreto, P. S. L. M., Longa, P., Naehrig, M., Ricardini, J. E., and Zanon, G. Sharper ring-lwe signatures. *IACR Cryptology ePrint Archive 2016* (2016), 1026.
12. Bellare, M., Poettering, B., and Stebila, D. From identification to signatures, tightly: A framework and generic transforms. In *ASIACRYPT* (2016).
13. Bellare, M., and Rogaway, P. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS* (1993).
14. Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., and Virza, M. Zerocash: Decentralized anonymous payments from bitcoin. In *IEEE SP* (2014).
15. Ben-Sasson, E., Chiesa, A., Genkin, D., Tromer, E., and Virza, M. Snarks for C: verifying program executions succinctly and in zero knowledge. In *CRYPTO* (2013).
16. Bernstein, D. J. Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete? `http://cr.yp.to/hash/collisioncost-20090823.pdf`.
17. Bernstein, D. J., Hopwood, D., Hülsing, A., Lange, T., Niederhagen, R., Papachristodoulou, L., Schneider, M., Schwabe, P., and Wilcox-O'Hearn, Z. SPHINCS: practical stateless hash-based signatures. In *EUROCRYPT* (2015).
18. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., and Zhandry, M. Random oracles in a quantum world. In *ASIACRYPT* (2011).
19. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E. B., Knezevic, M., Knudsen, L. R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S. S., and Yalçin, T. PRINCE - a low-latency block cipher for pervasive computing applications - extended abstract. In *ASIACRYPT* (2012).

20. Boyar, J., Matthews, P., and Peralta, R. Logic minimization techniques with applications to cryptology. *Journal of Cryptology 26*, 2 (2013), 280–312.
21. Brassard, G., Høyer, P., and Tapp, A. Quantum cryptanalysis of hash and claw-free functions. In *LATIN 1998* (Apr. 1998), C. L. Lucchesi and A. V. Moura, Eds., vol. 1380 of *LNCS*, Springer, Heidelberg, pp. 163–169.
22. Buchmann, J. A., Dahmen, E., and Hülsing, A. XMSS - A practical forward secure signature scheme based on minimal security assumptions. In *PQCrypto* (2011).
23. Campanelli, M., Gennaro, R., Goldfeder, S., and Nizzardo, L. Zero-knowledge contingent payments revisited: Attacks and payments for services. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017), ACM.
24. Canteaut, A., Carpov, S., Fontaine, C., Lepoint, T., Naya-Plasencia, M., Paillier, P., and Sirdey, R. Stream ciphers: A practical solution for efficient homomorphic-ciphertext compression. In *FSE* (2016).
25. Carlet, C., Goubin, L., Prouff, E., Quisquater, M., and Rivain, M. Higher-order masking schemes for s-boxes. In *FSE* (2012).
26. Costello, C., Fournet, C., Howell, J., Kohlweiss, M., Kreuter, B., Naehrig, M., Parno, B., and Zahur, S. Geppetto: Versatile verifiable computation. In *IEEE SP* (2015).
27. Courtois, N., Finiasz, M., and Sendrier, N. How to achieve a mceliece-based digital signature scheme. In *ASIACRYPT* (2001).
28. Cramer, R., Damgård, I., and Schoenmakers, B. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO* (1994).
29. Daemen, J., Peeters, M., Van Assche, G., and Rijmen, V. Nessie proposal: Noekeon. In *First Open NESSIE Workshop* (2000).
30. Dagdelen, Ö., Bansarkhani, R. E., Göpfert, F., Güneysu, T., Oder, T., Pöppelmann, T., Sánchez, A. H., and Schwabe, P. High-speed signatures from standard lattices. In *LATINCRYPT* (2014).
31. Dagdelen, Ö., Fischlin, M., and Gagliardoni, T. The fiat-shamir transformation in a quantum world. In *ASIACRYPT* (2013).
32. Dagdelen, Ö., Galindo, D., Véron, P., Alaoui, S. M. E. Y., and Cayrel, P. Extended security arguments for signature schemes. *Des. Codes Cryptography 78*, 2 (2016), 441–461.
33. De Cannière, C., and Preneel, B. Trivium. In *New Stream Cipher Designs - The eSTREAM Finalists*. 2008.
34. Derler, D., Orlandi, C., Ramacher, S., Rechberger, C., and Slamanig, D. Digital signatures from symmetric-key primitives. Cryptology ePrint Archive, Report 2016/1085, 2016. `http://eprint.iacr.org/2016/1085`.
35. Ducas, L. Accelerating bliss: the geometry of ternary polynomials. *IACR Cryptology ePrint Archive 2014* (2014).
36. Ducas, L., Durmus, A., Lepoint, T., and Lyubashevsky, V. Lattice signatures and bimodal gaussians. In *CRYPTO* (2013).
37. Ezerman, M. F., Lee, H. T., Ling, S., Nguyen, K., and Wang, H. A provably secure group signature scheme from code-based assumptions. In *Advances in Cryptology - ASIACRYPT* (2015), pp. 260–285.
38. Faugère, J., Gauthier-Umaña, V., Otmani, A., Perret, L., and Tillich, J. A distinguisher for high-rate mceliece cryptosystems. *IEEE Trans. Information Theory 59*, 10 (2013), 6830–6844.

39. Feo, L. D., Jao, D., and Plût, J. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *J. Mathematical Cryptology 8*, 3 (2014), 209–247.
40. Fiat, A., and Shamir, A. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO* (1986), pp. 186–194.
41. Galbraith, S. D., Petit, C., and Silva, J. Signature schemes based on supersingular isogeny problems. *IACR Cryptology ePrint Archive 2016* (2016), 1154.
42. Gennaro, R., Gentry, C., Parno, B., and Raykova, M. Quadratic span programs and succinct nizks without pcps. In *EUROCRYPT* (2013).
43. Gentry, C., Peikert, C., and Vaikuntanathan, V. Trapdoors for hard lattices and new cryptographic constructions. In *STOC* (2008).
44. Giacomelli, I., Madsen, J., and Orlandi, C. ZKBoo: Faster zero-knowledge for boolean circuits. In *USENIX Security* (2016).
45. Giacomelli, I., Madsen, J., and Orlandi, C. ZKBoo: Faster zero-knowledge for boolean circuits. Cryptology ePrint Archive, Report 2016/163, 2016. `http://eprint.iacr.org/2016/163`.
46. Goldfeder, S., Chase, M., and Zaverucha, G. Efficient post-quantum zero-knowledge and signatures. Cryptology ePrint Archive, Report 2016/1110, 2016. `http://eprint.iacr.org/2016/1110`.
47. Goldreich, O. Two remarks concerning the goldwasser-micali-rivest signature scheme. In *CRYPTO* (1986).
48. Goldreich, O., Micali, S., and Wigderson, A. How to prove all np-statements in zero-knowledge, and a methodology of cryptographic protocol design. In *CRYPTO* (1986).
49. Goldwasser, S., Micali, S., and Rackoff, C. The knowledge complexity of interactive proof-systems (extended abstract). In *STOC* (1985).
50. Grosso, V., Leurent, G., Standaert, F., and Varici, K. Ls-designs: Bitslice encryption for efficient masked software implementations. In *FSE* (2014).
51. Groth, J., and Sahai, A. Efficient Non-interactive Proof Systems for Bilinear Groups. In *EUROCRYPT* (2008).
52. Grover, L. K. A fast quantum mechanical algorithm for database search. In *STOC* (1996).
53. Güneysu, T., Lyubashevsky, V., and Pöppelmann, T. Practical lattice-based cryptography: A signature scheme for embedded systems. In *CHES* (2012).
54. Hellman, M. A cryptanalytic time-memory trade-off. *IEEE transactions on Information Theory 26*, 4 (1980), 401–406.
55. Hu, Z., Mohassel, P., and Rosulek, M. Efficient zero-knowledge proofs of non-algebraic statements with sublinear amortized cost. In *CRYPTO* (2015).
56. Hülsing, A., Rijneveld, J., Samardjiska, S., and Schwabe, P. From 5-pass mq-based identification to mq-based signatures. In *Cryptology ePrint Archive, Report 2016/708, to appear in Asiacrypt 2016* (2016).
57. Ishai, Y., Kushilevitz, E., Ostrovsky, R., and Sahai, A. Zero-knowledge proofs from secure multiparty computation. *SIAM Journal on Computing 39*, 3 (2009), 1121–1152.
58. Jawurek, M., Kerschbaum, F., and Orlandi, C. Zero-knowledge using garbled circuits: how to prove non-algebraic statements efficiently. In *ACM CCS* (2013).
59. Kaplan, M., Leurent, G., Leverrier, A., and Naya-Plasencia, M. Quantum Differential and Linear Cryptanalysis. *ArXiv e-prints* (Oct. 2015).
60. Kaplan, M., Leurent, G., Leverrier, A., and Naya-Plasencia, M. Breaking symmetric cryptosystems using quantum period finding. In *CRYPTO* (2016).

61. Katz, J. *Digital Signatures*. Springer, 2010.
62. Kiltz, E., Masny, D., and Pan, J. Optimal security proofs for signatures from identification schemes. In *CRYPTO* (2016).
63. Lamport, L. Constructing digital signatures from one-way functions. Tech. Rep. SRI-CSL-98, SRI Intl. Computer Science Laboratory, 1979.
64. Landais, G., and Sendrier, N. Cfs software implementation. Cryptology ePrint Archive, Report 2012/132, 2012.
65. Lyubashevsky, V. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT* (2009).
66. Lyubashevsky, V. Lattice signatures without trapdoors. In *EUROCRYPT* (2012).
67. McEliece, R. J. A public-key cryptosystem based on algebraic coding theory. Tech. Rep. DSN PR 42-44, 1978.
68. McGrew, D. A., Kampanakis, P., Fluhrer, S. R., Gazdag, S., Butin, D., and Buchmann, J. A. State management for hash-based signatures. In *Security Standardisation Research* (2016).
69. Méaux, P., Journault, A., Standaert, F., and Carlet, C. Towards stream ciphers for efficient FHE with low-noise ciphertexts. In *EUROCRYPT* (2016).
70. Melchor, C. A., Gaborit, P., and Schrek, J. A new zero-knowledge code based identification scheme with reduced communication. In *ITW* (2011).
71. Merkle, R. C. A certified digital signature. In *CRYPTO* (1989).
72. Niederreiter, H. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory* (1986).
73. Ohta, K., and Okamoto, T. On concrete security treatment of signatures derived from identification. In *CRYPTO* (1998).
74. Patarin, J., Courtois, N., and Goubin, L. Quartz, 128-bit long digital signatures. In *CT-RSA* (2001).
75. Peikert, C. A decade of lattice cryptography. *Foundations and Trends in Theoretical Computer Science 10*, 4 (2016).
76. Petzoldt, A., Chen, M., Yang, B., Tao, C., and Ding, J. Design principles for hfev- based multivariate signature schemes. In *ASIACRYPT* (2015).
77. Pointcheval, D., and Stern, J. Security proofs for signature schemes. In *EUROCRYPT* (1996).
78. S. Ames, C. Hazay, Y. I., and Venkitasubramaniam, M. Ligero: Lightweight sublinear arguments without a trusted setup. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017), ACM.
79. Sakumoto, K., Shirai, T., and Hiwatari, H. Public-key identification schemes based on multivariate quadratic polynomials. In *CRYPTO* (2011).
80. Schnorr, C. Efficient signature generation by smart cards. *J. Cryptology 4*, 3 (1991).
81. Shor, P. W. Polynominal time algorithms for discrete logarithms and factoring on a quantum computer. In *ANTS-I* (1994).
82. Stern, J. A new identification scheme based on syndrome decoding. In *CRYPTO* (1993).
83. Unruh, D. Quantum proofs of knowledge. In *EUROCRYPT 2012* (Apr. 2012), D. Pointcheval and T. Johansson, Eds., vol. 7237 of *LNCS*, Springer, Heidelberg, pp. 135–152.
84. Unruh, D. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *EUROCRYPT 2015, Part II* (Apr. 2015), E. Oswald and M. Fischlin, Eds., vol. 9057 of *LNCS*, Springer, Heidelberg, pp. 755–784.

85. Unruh, D. Computationally binding quantum commitments. In *EUROCRYPT* (2016).
86. Véron, P. Improved identification schemes based on error-correcting codes. *Appl. Algebra Eng. Commun. Comput. 8*, 1 (1996).
87. Yoo, Y., Azarderakhsh, R., Jalali, A., Jao, D., and Soukharev, V. A post-quantum digital signature scheme based on supersingular isogenies. Cryptology ePrint Archive, Report 2017/186, 2017. `http://eprint.iacr.org/2017/186`.

A Additional Material on ZKBoo

In Scheme 3 we restate the full ZKBoo protocol.

For public ϕ and $y \in L_\phi$, the prover has x, such that $y = \phi(x)$. $\texttt{Com}(\cdot)$ is a secure commitment scheme. The prover and verifier use the hash function $H(\cdot)$, which is modeled as random oracle. The integer t is the number of parallel iterations.

$p \leftarrow \texttt{Prove}(x)$:

1. For each iteration r_i, $i \in [1,t]$: Sample random tapes $k_1^{(i)}, k_2^{(i)}, k_3^{(i)}$ and run the decomposition to get an output view $\mathsf{View}_j^{(i)}$ and output share $y_j^{(i)}$. In particular, for each player P_j:

$$\begin{aligned} (x_1^{(i)}, x_2^{(i)}, x_3^{(i)}) &\leftarrow \texttt{Share}(x, k_1^{(i)}, k_2^{(i)}, k_3^{(i)}), \\ \mathsf{View}_j^{(i)} &\leftarrow \texttt{Update}(\texttt{Update}(\cdots\texttt{Update}(x_j^{(i)}, x_{j+1}^{(i)}, k_j^{(i)}, k_{j+1}^{(i)})\ldots)\ldots)\ldots), \\ y_j^{(i)} &\leftarrow \texttt{Output}(\mathsf{View}_j^{(i)}) \end{aligned}$$

   Commit $[C_j^{(i)}, D_j^{(i)}] \leftarrow \mathsf{Com}(k_j^{(i)}, \mathsf{View}_j^{(i)})$ and let $a^{(i)} = (y_1^{(i)}, y_2^{(i)}, y_3^{(i)}, C_1^{(i)}, C_2^{(i)}, C_3^{(i)})$.
2. Compute the challenge: $e \leftarrow H(a^{(1)}, \ldots, a^{(t)})$. Interpret the challenge such that for $i \in [1,t]$, $e^{(i)} \in \{1,2,3\}$
3. For each iteration $r_i, i \in [1,t]$, let $z^{(i)} = (D_e^{(i)}, D_{e+1}^{(i)})$.
4. Output $p = [(a^{(1)}, z^{(1)}), (a^{(2)}, z^{(2)}), \cdots, (a^{(t)}, z^{(t)})]$

$b \leftarrow \texttt{Verify}(y, p)$:

1. Compute the challenge: $e' \leftarrow H(a^{(1)}, \cdots, a^{(t)})$. Interpret the challenge such that for $i \in [1,t]$, $e'^{(i)} \in \{1,2,3\}$.
2. For each iteration r_i, $i \in [1,t]$: If there exists $j \in \{e'^{(i)}, e'^{(i)}+1\}$ such that $\mathsf{Open}(C_j^{(i)}, D_j^{(i)}) = \perp$, output Reject. Otherwise, for all $j \in \{e'^{(i)}, e'^{(i)}+1\}$, set $\{k_j^{(i)}, \mathsf{View}_j^{(i)}\} \leftarrow \mathsf{Open}(C_j^{(i)}, D_j^{(i)})$.
3. If $\texttt{Reconstruct}(y_1^{(i)}, y_2^{(i)}, y_3^{(i)}) \neq y$, output Reject. If there exists $j \in \{e'^{(i)}, e'^{(i)}+1\}$ such that $y_j^{(i)} \neq \texttt{Output}(\mathsf{View}_j^{(i)})$, output Reject. For each wire value $w_j^{(e)} \in \mathsf{View}_e$, if $w_j^{(e)} \neq \texttt{Update}(\mathsf{view}_e^{(j-1)}, \mathsf{view}_{e+1}^{(j-1)}, k_e, k_{e+1})$ output Reject.
4. Output Accept.

Scheme 3: The ZKBoo non-interactive proof system

A.1 (2,3)-Decomposition

We define the experiment $\mathsf{EXP}^{(\phi,\mathsf{x})}_{\mathsf{decomp}}$ in Scheme 4, which runs the decomposition over a circuit ϕ on input x: We say that $\mathcal{D}$ is a $(2,3)$-*decomposition* of ϕ if the following two properties hold when running $\mathsf{EXP}^{(\phi,\mathsf{x})}_{\mathsf{decomp}}$:

$\mathsf{EXP}^{(\phi,\mathsf{x})}_{\mathsf{decomp}}$:

1. First run the Share function on x: $\mathsf{view}_1^{(0)}, \mathsf{view}_2^{(0)}, \mathsf{view}_3^{(0)} \leftarrow \mathtt{Share}(x, k_1, k_2, k_3)$
2. For each of the three views, call the update function successively for every gate in the circuit: $\mathsf{view}_i^{(j)} = \mathtt{Update}(\mathsf{view}_i^{(j-1)}, \mathsf{view}_{i+1}^{(j-1)}, k_i, k_{i+1})$ for $i \in [1,3]$, $j \in [1,n]$
3. From the final views, compute the output share of each view: $y_i \leftarrow \mathtt{Output}(\mathsf{View}_i)$

Scheme 4: Decomposition Experiment

(Correctness) For all circuits ϕ, for all inputs x and for the y_i's produced by , for all circuits ϕ, for all inputs x,

$$\Pr[\phi(x) = \mathtt{Reconstruct}(y_1, y_2, y_3)] = 1$$

(2-Privacy) Let $\mathcal{D}$ be correct. Then for all $e \in \{1,2,3\}$ there exists a PPT simulator $\mathcal{S}_e$ such that for any probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$, for all circuits ϕ, for all inputs x, and for the distribution of views and k_i's produced by $\mathsf{EXP}^{(\phi,\mathsf{x})}_{\mathsf{decomp}}$ we have that $\big|\Pr[\mathcal{A}(x, y, k_e, \mathsf{View}_e, k_{e+1}, \mathsf{View}_{e+1}, y_{e+2}) = 1] - \Pr[\mathcal{A}(x, y, \mathcal{S}_e(\phi, y)) = 1]\big|$ is negligible.

A.2 Linear Decomposition of a Circuit

ZKBoo uses an explicit (2,3)-decomposition, which we recall here. Let R be an arbitrary finite ring and ϕ a function such that $\phi : R^m \rightarrow R^\ell$ can be expressed by an n-gate arithmetic circuit over the ring using addition by constant, multiplication by constant, binary addition and binary multiplication gates. A $(2,3)-$decomposition of ϕ is given by the following functions. In the notation below, arithmetic operations are done in R^s where the operands are elements of R^s):

- $(x_1, x_2, x_3) \leftarrow \mathtt{Share}(x, k_1, k_2, k_3)$ samples random $x_1, x_2, x_3 \in R^m$ such that $x_1 + x_2 + x_3 = x$.
- $y_i \leftarrow \mathtt{Output}_i(\mathsf{view}_i^{(n)})$ selects the ℓ output wires of the circuit as stored in the view $\mathsf{view}_i^{(n)}$.
- $y \leftarrow \mathtt{Reconstruct}(y_1, y_2, y_3) = y_1 + y_2 + y_3$
- $\mathsf{view}_i^{(j+1)} \leftarrow \mathtt{Update}_i^{(j)}(\mathsf{view}_i^{(j)}, \mathsf{view}_{i+1}^{(j)}, k_i, k_{i+1})$ computes P_i's view of the output wire of gate g_j and appends it to the view. Notice that it takes as input the views and random tapes of both party P_i as well as party P_{i+1}. We use w_k to refer to the k-th wire, and we use $w_k^{(i)}$ to refer to the value of w_k in party P_i's view. The update operation depends on the type of gate g_j.

The gate-specific operations are defined as follows.

Addition by Constant ***(*** $w_b = w_a + k$ ***).***

$$w_b^{(i)} = \begin{cases} w_a^{(i)} + k & \text{if } i = 1, \\ w_a^{(i)} & \text{otherwise.} \end{cases}$$

Multiplication by Constant ***(*** $w_b = w_a \cdot k$ ***).***

$$w_b^{(i)} = k \cdot w_a^{(i)}$$

Binary Addition ***(*** $w_c = w_a + w_b$ ***).***

$$w_c^{(i)} = w_a^{(i)} + w_b^{(i)}$$

Binary Multiplication ***(*** $w_c = w_a \cdot w_b$ ***).***

$$\begin{aligned} w_c^{(i)} = \; & w_a^{(i)} \cdot w_b^{(i)} + w_a^{(i+1)} \cdot w_b^{(i)} + \\ & w_a^{(i)} \cdot w_b^{(i+1)} + R_i(c) - R_{i+1}(c), \end{aligned}$$

where $R_i(c)$ is the c-th output of a pseudorandom generator seeded with k_i.

Note that with the exception of the constant addition gate, the gates are symmetric for all players. Also note that P_i can compute all gate types locally with the exception of binary multiplication gates as this requires inputs from P_{i+1}. In other words, for every operation except binary multiplication, the `Update` function does not use the inputs from the second party, i.e., $\mathsf{view}_{i+1}^{(j)}$ and k_{i+1}.

While we do not give the details here, [45] shows that this decomposition meets the correctness and 2-privacy requirements of Definition 1.

B Description of LowMC

LowMC by Albrecht et al. [6,4] is very parameterizable symmetric encryption scheme design enabling instantiation with low AND depth and low multiplicative complexity. Given any blocksize, a choice for the number of S-boxes per round, and security expectations in terms of time and data complexity, instantiations can be created minimizing the AND depth, the number of ANDs, or the number of ANDs per encrypted bit. Table 2 lists the choices for the parameters which are also highlighted in the figures.

The description of LowMC is possible independently of the choice of parameters using a partial specification of the S-box and arithmetic in vector spaces over $\mathbb{F}_2$. In particular, let n be the blocksize, m be the number of S-boxes, k the key size, and r the number of rounds, we choose round constants $C_i \xleftarrow{R} \mathbb{F}_2^n$ for $i \in [1, r]$, full rank matrices $K_i \xleftarrow{R} \mathbb{F}_2^{n \times k}$ and regular matrices $L_i \xleftarrow{R} \mathbb{F}_2^{n \times n}$ independently during the instance generation and keep them fixed. Keys for LowMC are generated by sampling from $\mathbb{F}_2^k$ uniformly at random.

Blocksize	S-boxes	Keysize	Rounds
n	m	k	r
256	1	256	316
256	10	256	38
256	42	256	14

Table 2. A range of different parameter sets for LowMC. All parameters are computed for data complexity $d = 1$

LowMC encryption starts with key whitening which is followed by several rounds of encryption. A single round of LowMC is composed of an S-box layer, a linear layer, addition with constants and addition of the round key, i.e.

$$\begin{aligned}\textsc{LowMCRound}(i) &= \textsc{KeyAddition}(i)\\ &\circ \textsc{ConstantAddition}(i)\\ &\circ \textsc{LinearLayer}(i) \circ \textsc{SboxLayer}.\end{aligned}$$

SboxLayer is an m-fold parallel application of the same 3-bit S-box on the first $3 \cdot m$ bits of the state. The S-box is defined as $S(a, b, c) = (a \oplus bc, a \oplus b \oplus ac, a \oplus b \oplus c \oplus ab)$.

The other layers only consist of $\mathbb{F}_2$-vector space arithmetic. LinearLayer(i) multiplies the state with the linear layer matrix L_i, ConstantAdditon(i) adds the round constant C_i to the state, and KeyAddition(i) adds the round key to the state, where the round key is generated by multiplying the master key with the key matrix K_i.

Algorithm 1 gives a full description of the encryption algorithm.

Algorithm 1 LowMC encryption for key matrices $K_i \in \mathbb{F}_2^{n\times k}$ for $i \in [0, r]$, linear layer matrices $L_i \in \mathbb{F}_2^{n\times n}$ and round constants $C_i \in \mathbb{F}_2^n$ for $i \in [1, r]$.

```
Require: plaintext $p \in \mathbb{F}_2^n$ and key $y \in \mathbb{F}_2^k$
  $s \leftarrow K_0 \cdot y + p$
  for $i \in [1, r]$ do
      $s \leftarrow Sbox(s)$
      $s \leftarrow L_i \cdot s$
      $s \leftarrow C_i + s$
      $s \leftarrow K_i \cdot y + s$
  end for
  return $s$
```

C Building Blocks

Commitments. Formally a (non-interactive) commitment scheme consists of three algorithms KG, Com, Open with the following properties:

$\mathsf{KG}(1^\kappa)$: The key generation algorithm, on input the security parameter κ it outputs a public key pk (we henceforth assume pk to be an implicit input to the subsequent algorithms).

$\mathsf{Com}(M)$: On input of a message M, the commitment algorithm outputs $(C(M), D(M)) \leftarrow \mathsf{Com}(M; R)$, where R are the coin tosses. $C(M)$ is the commitment string, while $D(M)$ is the decommitment string which is kept secret until opening time.

$\mathsf{Open}(C, D)$: On input C, D, the verification algorithm either outputs a message M or $\perp$.

We note that if the sender refuses to open a commitment we can set $D = \perp$ and $\mathsf{Open}(\mathsf{pk}, C, \perp) = \perp$. Computationally secure commitments must satisfy the following properties

Correctness If $(C(M), D(M)) = \mathsf{Com}(M)$ then $\mathsf{Open}(\mathsf{pk}, C(M), D(M)) = M$.

Hiding For every message pair M, M' the probability ensembles $\{C(M)\}_{\kappa\in\mathbb{N}}$ and $\{C(M')\}_{\kappa\in\mathbb{N}}$ are computationally indistinguishable for security parameter κ.

Binding We say that an adversary $\mathcal{A}$ wins if it outputs C, D, D' such that $\mathsf{Open}(C, D) = M$, $\mathsf{Open}(C, D') = M'$ and $M \neq M'$. We require that for all efficient algorithms $\mathcal{A}$, the probability that $\mathcal{A}$ wins is negligible in the security parameter.

To simplify our notation, we will often not explicitly generate the public key pk when we make use of commitments.

Our implementation uses hash-based commitments, which requires modeling the hash function as a random oracle in our security analysis. Note also that randomizing the Com function may not be necessary if M has high entropy.

One-Way Functions. We define the notion of families of one-way functions.

Definition 2. *A family of functions $\{f_k\}_{k\in\mathsf{K}_\kappa}$ with $f_k : \mathsf{D}_\kappa \to \mathsf{R}_\kappa$ is called one-way, if (1) for all κ and for all $k \in \mathsf{K}_\kappa$ there exists a PPT algorithm $\mathcal{A}_1$ so that $\forall x \in \mathsf{D}_\kappa : \mathcal{A}_1(x) = f_k(x)$, and (2) for every PPT algorithm $\mathcal{A}_2$ there is a negligible function $\epsilon(\cdot)$ such that it holds that*

$$\Pr\left[k \xleftarrow{R} \mathsf{K}_\kappa, x \xleftarrow{R} \mathsf{D}_\kappa, x^* \leftarrow \mathcal{A}_2(1^\kappa, f_k(x)) \;:\; f(x) = f(x^*)\right] \leq \epsilon(\kappa).$$

Pseudorandom Functions and Generators. We require the notion of pseudorandom functions and generators, which we formally recall below.

Definition 3 (Pseudorandom Function). *Let $\{f_k\}_{k\in\mathsf{K}_\kappa}$ be an efficiently computable, length-preserving function family with $f_k : \{0,1\}^\kappa \to \{0,1\}^\kappa$. We say that $\{f_k\}$ is a* pseudorandom function *(PRF) family, if for all PPT distinguishers D,*

$$|\Pr[D^{f_k}(1^\kappa) = 1] - \Pr[D^{F_\kappa}(1^\kappa) = 1]|$$

is negligible in κ, where $k \leftarrow \{0,1\}^\kappa$ is chosen uniformly at random and F_κ is chosen uniformly at random from the set of functions mapping κ-bit strings to κ-bit strings.

We now define a weaker notion of a pseudorandom function in which we put an upper bound on the number of queries that the distinguisher can make to its oracle.

Definition 4 (q-Pseudorandom Function). *Let $\{f_k\}$ and F_κ be as defined in Definition 3, and let q be a positive integer constant. We say that F is a* q-pseudorandom function *(q-PRF) if for all PPT distinguishers D that make at most q queries to their oracle,*

$$|\Pr[D^{f_k}(1^\kappa) = 1] - \Pr[D^{F_\kappa}(1^\kappa) = 1]|$$

is negligible in κ.

Note that a pseudorandom function is also a q-pseudorandom function for any constant q. When considering concrete security of PRFs against quantum attacks, we assume that an n-bit function provides $n/2$ bits of security.

Pseudorandom Generators. We require the notion of pseudorandom generators, which we formally recall below.

Definition 5 (Pseudorandom Generator). *An (n, ℓ)* pseudorandom generator *(PRG) is a function $P : \{0,1\}^n \to \{0,1\}^\ell$ that expands an n-bit seed to an ℓ-bit random string. Informally, the PRG is said to be* secure *if for randomly chosen seeds, the output is indistinguishable from the uniform distribution on $\{0,1\}^\ell$.*

Concretely, we assume that AES-256 in counter mode provides 128 bits of PRG security, when used to expand 256-bit seeds to outputs less than 1kB in length.

Signature Schemes. Below we recall a standard definition of signature schemes.

Definition 6. *A signature scheme Σ is a triple* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *of PPT algorithms, which are defined as follows:*

$\mathsf{Gen}(1^\kappa)$: *This algorithm takes a security parameter κ as input and outputs a secret (signing) key* sk *and a public (verification) key* pk *with associated message space $\mathcal{M}$ (we may omit to make the message space $\mathcal{M}$ explicit).*

$\mathsf{Sign}(\mathsf{sk}, m)$: *This algorithm takes a secret key* sk *and a message $m \in \mathcal{M}$ as input and outputs a signature σ.*

$\mathsf{Verify}(\mathsf{pk}, m, \sigma)$: *This algorithm takes a public key* pk*, a message $m \in \mathcal{M}$ and a signature σ as input and outputs a bit $b \in \{0,1\}$.*

Besides the usual correctness property, Σ needs to provide some unforgeability notion. In this paper we are only interested in schemes that provide existential unforgeability under adaptively chosen message attacks ($\mathsf{EUF\text{-}CMA}$ security), which we define below.

Definition 7 ($\mathsf{EUF\text{-}CMA}$). *A signature scheme Σ is* $\mathsf{EUF\text{-}CMA}$ *secure, if for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that*

$$\Pr\Big[(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^\kappa),\ (m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk}, \cdot)}(\mathsf{pk}) :$$
$$\mathsf{Verify}(\mathsf{pk}, m^*, \sigma^*) = 1 \ \wedge\ m^* \notin \mathcal{Q}^{\mathsf{Sign}}\Big] \leq \varepsilon(\kappa),$$

where the environment keeps track of the queries to the signing oracle via $\mathcal{Q}^{\mathsf{Sign}}$.

D Security of Key Generation

In this section, we argue that using the block cipher in the way we use it can serve as our hard instance generator to generate keys for our signature scheme. Below, we recall the definition of hard instance generators as presented in [84]:

Definition 8 (Hard Instance Generators). *An algorithm G is called a* **hard instance generator** *for a relation R if*

1. *there is a negligible function $\epsilon_1(\cdot)$ so that it holds that*

$$\Pr[(y,x) \leftarrow G(1^\kappa) : (y,x) \in R] \geq 1 - \epsilon_1(\kappa),$$

2. *and for every PPT algorithm A there is a negligible function $\epsilon_2(\cdot)$ so that it holds that*

$$Pr[(y,x) \leftarrow G(1^\kappa), x^* \leftarrow A(y) : (y,x^*) \in R] \leq \epsilon_2(\kappa).$$

To establish a relation between public and secret keys, we use a family of block-cipher-based one-way functions $\{f_k\}$ where

$$f_k(x) := \mathsf{Enc}(x,k).$$

That is, $\mathsf{Enc}(x,k)$ denotes the encryption of a single block $k \in \{0,1\}^{c\cdot\kappa}$ with respect to key $x \in \{0,1\}^{c\cdot\kappa}$. Upon key generation, one first samples a concrete one-way function $\{f_k\}$ with respect to security parameter κ uniformly at random by sampling a uniformly random block $k \in \{0,1\}^{c\cdot\kappa}$. This function is then fixed by including k in the public key of the scheme, which implicitly defines the relation R. That is

$$(y,x) \in R \Longleftrightarrow y = f_k(x).$$

Now, we assume that using LowMC in this way yields a suitable one-way function. As already mentioned by Unruh in [84], a one-way function directly yields a suitable hard instance generator (observe the similarity in the definitions). To make our results more general, we show that any block-cipher where the keysize is equal to the blocksize, and in particular equal to $c\cdot\kappa$ (where we set $c = 1$ in the classical setting, whereas we set $c = 2$ in the post-quantum setting to account for the generic speedup imposed by Grover's algorithm [52]), when viewed as a family of PRFs, also yields a suitable one-way function family.

Theorem 1. *Let $\{f_k\}_{k\in\mathsf{K}_\kappa}$ with $f_k : \mathsf{M}_\kappa \rightarrow \mathsf{M}_\kappa$ be a family of pseudorandom functions, then $\{f_k\}_{k\in\mathsf{K}_\kappa}$ is also a family of one-way functions with respect to k for any input $x \in \mathsf{M}_\kappa$.*

Proof. The first condition of Definition 2 clearly holds, as for any κ, any $k \in \mathsf{K}_\kappa$, and any $x \in \mathsf{M}_\kappa$ we can efficiently compute $y = f_k(x)$ by definition 3. What remains is to prove the second condition. In particular, we need to show that for every PPT algorithm $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ so that it holds that

$$\Pr[k \overset{R}{\leftarrow} \mathsf{K}_\kappa, k^* \leftarrow \mathcal{A}(1^\kappa, f_k(x)) : f_k(x) = f_{k^*}(x)] \leq \epsilon(\kappa),$$

for any $x \in \mathsf{M}_\kappa$. We denote by $\texttt{keyset}(y)$ the set of keys $\mathcal{B}$ such that for all $k \in \mathcal{B},\ f_k(r) = y$. If there is no such satisfying key, $\texttt{keyset}(y)$ returns the empty set. For an algorithm $\mathcal{A}$, denote by t_y the probability that $\mathcal{A}$ will output a key k' on input y such that $f_{k'}(r) = y$. Then, using this notation we can rewrite the probability above as

$$P_1 := \sum_y \frac{|\texttt{keyset}(y)|}{|\mathsf{K}_\kappa|} \cdot t_y$$

and we need to show that P_1 is negligible for any $\mathcal{A}$.

First, we define, probability P_2, which is the probability that $\mathcal{A}$ will output the "correct key", by which we mean the same key that was chosen to generate y. Since the key was chosen uniformly at random, information-theoretically, there is no way for $\mathcal{A}$ to distinguish between the "correct key" and any other valid key (i.e. any k' for which $f_{k'}(r) = y$). Thus, the only strategy that $\mathcal{A}$ has is to output any valid key and with probability $1/\texttt{keyset}(y)$, the key that it outputs will be the "correct key". Thus, we have:

$$\begin{aligned} P_2 :&= \Pr[k \overset{R}{\leftarrow} \mathsf{K}_\kappa, k^* \leftarrow \mathcal{A}(1^\kappa, f_k(x)) : k = k^*] \\ &= \sum_y \frac{|\texttt{keyset}(y)|}{|\mathsf{K}_\kappa|} \cdot t_y \cdot \frac{1}{|\texttt{keyset}(y)|} \\ &= \frac{1}{|\mathsf{K}_\kappa|} \sum_y t_y \end{aligned}$$

We now show that P_2 is negligible. Assume that there exists an $\mathcal{A}$ for which P_2 is equal to a non-negligible ϵ. Then we can build a distinguisher D that distinguishes between F and a random function as follows:

```
$D^{\mathcal{O}}(1^{|k|})$

1. $y' \leftarrow \mathcal{O}(r)$. Queries the oracle on $x$ and receive response $y'$.
2. Invoke $\mathcal{A}$ on input $y'$.
   (a) if $\perp \leftarrow \mathcal{A}(y')$, output 0.
   (b) if $k^* \leftarrow \mathcal{A}(y')$, check that this is the "correct key" as follows
       i. First check that $f_{k^*}(r) = y$. If not, output 0. Else, continue
       ii. Next, choose a value $q \leftarrow \mathsf{M}_\kappa$ uniformly at random and query on
           that value – i.e. query for $z \leftarrow \mathcal{O}(q)$.
       iii. Check that $z = f_{k^*}(q)$. If it does not, output 0. If it does, output 1.
```

Now, let's analyze the output of D. Whenever $\mathcal{A}$ outputs the "correct key", D will output 1. Moreover, $\mathcal{A}$ will output the correct key with probability ϵ. Thus, if D's oracle is a pseudorandom function – i.e. if $\mathcal{O} = f_k$, then with probability at least ϵ, D will output 1. To see that this is true notice that when $\mathcal{O} = f_k$, the key k for f_k is chosen from the same distribution as in a real execution, and thus $\mathcal{A}$'s success probability on outputting the "correct key" will be exactly ϵ.[24]

If, however, $\mathcal{O}$ is a random function – i.e. $\mathcal{O} = F_n$, then D will only output 1 in the event that $F_n(q) = f_{k^*}(q)$. In step (iii), once we have chosen a key k^*, the probability of the random function agreeing with F_{k^*} on q is $\delta = \frac{1}{|\mathcal{M}|-1}$, which is negligible in $|k|$ since $|\mathcal{M}| \geq |\mathcal{K}| = 2^{|k|}$.

Thus, we have built a good distinguisher since:

$$|\Pr[D^{f_k}(1^{|k|}) = 1] - \Pr[D^{F_n}(1^{|k|}) = 1]| \geq \epsilon - \delta$$

which is non-negligible.

This contradicts our assumption that $\{f_k\}$ is a pseudorandom function family, and we therefore conclude that P_2 is negligible.

We now show that $|P_1 - P_2|$ is negligible. Once again, consider an algorithm $\mathcal{A}$ that on input y outputs a key k' such that $f_{k'}(x) = y$ with probability t_y. Consider the following two games.

Game 1. A key $k \leftarrow \mathsf{K}_\kappa$ is chosen uniformly at random and $y = f_k(x)$ is given to the adversary. The adversary wins if it can produce a key k' such that $f_{k'}(x) = y$. The probability of $\mathcal{A}$ succeeding at this game is exactly P_1:

$$\sum_y \frac{|\texttt{keyset}(y)|}{|\mathsf{K}_\kappa|} \cdot t_y$$

Game 2. $y \leftarrow \mathcal{M}$ is chosen uniformly at random and given to the adversary. The adversary wins if it can output a key k' such that $F_{k'}(r) = y$. The difference between this game and the previous one is that now we choose y uniformly irrespective of the keys. Thus all y's will be chosen with equal probability no matter how many keys (if any) map r to y. The success probability of $\mathcal{A}$ in this game is

$$P_3 := \frac{1}{|\mathsf{M}_\kappa|} \sum_y t_y$$

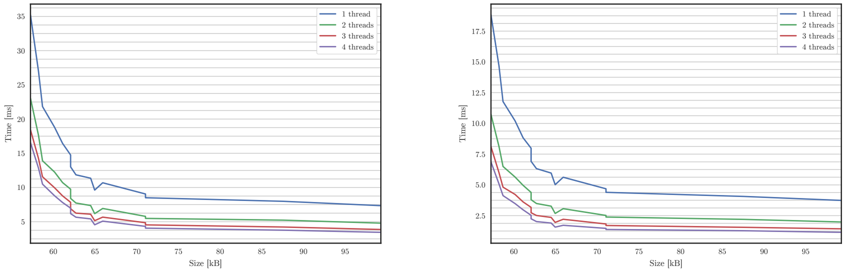
Now, if you could distinguish between Game 1 and Game 2, you could build an algorithm D that distinguishes f from a random function. D simply queries its oracle at r, and send the response y to A. If the oracle is a pseudorandom function, then the success probability will be exactly the same as Game 1, namely P_1. If it is a random function, the success probability is exactly the same as Game 2, namely P_3. Thus, by Definition 3, we know that $|P_1 - P_3|$ is negligible.

[24] It is possible that when $\mathcal{O} = f_k$, D will output 1 with probability greater than ϵ –i.e. if $\mathcal{A}$ outputs the wrong key that happens to agree with the "correct key" on the queried values, but for the sake of our argument it suffices to show that it outputs 1 with probability at least ϵ.

Since $|\mathsf{K}_\kappa| \leq |\mathsf{M}_\kappa|$, then $P_2 \leq P_3$, and in particular, when $|\mathsf{K}_\kappa| = |\mathsf{M}_\kappa|$, $P_2 = P_3$. We thus have that $|P_1 - P_2|$ is negligible. Since we have shown that both P_2 and $|P_1 - P_2|$ are negligible, it follows that P_1 is negligible as well.□

E Parallelization of Proofs

One positive aspect regarding the t parallel repetitions is that they are independent of each other. This observation was also made for ZKBoo in [44]. In particular, this holds for all steps in the signing and verification algorithm up to the initial requests to OpenSSL's random number generator and the computation of the challenge. This allows us to take advantage of the multi-core architecture of modern processors using OpenMP.[25] As exemplified for Fish in Figure 2, we can observe a significant performance increase until the number of threads matches the actual number of CPU cores[26]. We note that exactly the same effects also occur for instantiations of Picnic. Furthermore, they also occur regardless of the LowMC parameters. The speed-up is not linear with our current implementation. The speed-up from one to two threads is about 2x, but becomes smaller as additional cores are added, likely because memory access becomes a bottleneck.



(a) Sign

(b) Verify

Fig. 2. Runtime of the parallelized version of Sign and Verify of Fish using an increasing number of threads. The x-axis shows the running time, while y-axis shows the various LowMC parameter sets, sorted by signature size (as in Figure 1).

[25] http://openmp.org

[26] HyperThreading was disabled to reduce noise in the benchmarks.

F Security of the proof system in the quantum random oracle model

Here we prove that the proof system we get by applying our modified Unruh transform to ZKB++ as described in Section 5 is both zero knowledge and simulation-extractable in the quantum random oracle model.

Before we begin, we note that the quantum random oracle model is highly non-trivial, and a lot of the techniques used in standard random oracle proofs do not apply. The adversary is a quantum algorithm that may query the oracle on quantum inputs which are a superposition of states and receive superposition of outputs. If we try to measure those states, we change the outcome, so we do not for example have the same ability to view the adversary's input and program the responses that we would in the standard ROM.

Here we rely on lemmas from Unruh's work on quantum-secure Fiat-Shamir like proofs [84]. We follow his proof strategy as closely as possible, modifying it to account for the optimizations we made and the fact that we have only 3-special soundness in our underlying Σ-protocol.

Zero Knowledge This proof very closely follows the proof from [84]. The main difference is that we also use the random oracle to form our commitments, which is addressed in the transition from game 2 to game 3 below.

Consider the simulator described in Figure 5. From this point on we assume for simplicity of notation that View_3 includes x_3.

We proceed via a series of games.

Game 1: This is the real game in the quantum random oracle model. Let H_{com} be the random oracle used for forming the commitments, H_{chal} be the random oracle used for forming the challenge, and G be the additional random permutation.

Game 2: We change the prover so that it first chooses random $e* = e*^{(1)}, \ldots, e*^{(t)}$, and then on step 2, it programs $H_{chal}(a^{(1)}, \ldots, a^{(t)}, h^{(1)}, \ldots, h^{(t)}) = e*$. Note that each the $a^{(1)}, \ldots, a^{(t)}, h^{(1)}, \ldots, h^{(t)}$ has sufficient collision-entropy, since it includes $\{h^{(i)} = (g_1^{(i)}, g_2^{(i)}, g_3^{(i)})\}$, the output of a permutation on input whose first k bits are chosen at random (the $k_j^{(i)}$), so we can apply Corollary 11 from [84] (using a hybrid argument) to argue that Game 1 and Game 2 are indistinguishable.

Game 3: We replace the output of each $H_{com}(k_{e*^{(i)}}, \mathsf{View}_{e*^{(i)}})$ and $G(k_{e*^{(i)}}, \mathsf{View}_{e*^{(i)}})$ with a pair of random values.

First, note that H_{com} and G are always called (by the honest party) on the same inputs, so we will consider them as a single random oracle with a longer output space, which we refer to as H for this proof.

Now, to show that Games 2 and 3 are indistinguishable, we proceed via a series of hybrids, where the i-th hybrid replaces the first i such outputs with random values.

To show that the i-th and $i+1$-st hybrid are indistinguishable, we rely on Lemma 9 from [84]. This lemma says the following: For any quantum A_0, A_1

$p \leftarrow \texttt{Sim}(x)$: In the simulator, we follow Unruh, and replace the initial state (before programming) of the random oracles with random polynomials of degree $2q-1$ where q is an upper bound on the number of queries the adversary makes.

1. For $i \in [1,t]$, choose random $e^{(i)} \leftarrow \{1,2,3\}$. Let e be the corresponding binary string.
2. For each iteration $r_i, i \in [1,t]$: Sample random seeds $k^{(i)}_{e^{(i)}}, k^{(i)}_{e^{(i)}+1}$ and run the circuit decomposition simulator to generate $\mathsf{View}^{(i)}_{e^{(i)}}$, $\mathsf{View}^{(i)}_{e^{(i)}+1}$, output shares $y^{(i)}_1$, $y^{(i)}_2$, $y^{(i)}_3$, and if $e^{(i)}=1$ $x^{(i)}_3$.
   For $j = e^{(i)}, e^{(i)}+1$ commit $[C^{(i)}_j, D^{(i)}_j] \leftarrow [H(k^{(i)}_j, \mathsf{View}^{(i)}_j), k^{(i)}_j || \mathsf{View}^{(i)}_j]$, and compute $g^{(i)}_j = G(k^{(i)}_j, \mathsf{View}^{(i)}_j)$.
   Choose random $C_{e^{(i)}+2}, g^{(i)}_{e^{(i)}}$
   Let $a^{(i)} = (y^{(i)}_1, y^{(i)}_2, y^{(i)}_3, C^{(i)}_1, C^{(i)}_2, C^{(i)}_3)$. And $h^{(i)} = g^{(i)}_1, g^{(i)}_2, g^{(i)}_3$.
2. Set the challenge: program $H(a^{(1)}, \ldots, a^{(t)}) := e$.
3. For each iteration r_i, $i \in [1,t]$: let $b^{(i)} = (y^{(i)}_{e^{(i)}+2}, C^{(i)}_{e^{(i)}+2})$ and set

$$z^{(i)} \leftarrow \begin{cases} (\mathsf{View}^{(i)}_2, k^{(i)}_1, k^{(i)}_2) & \text{if } e^{(i)} = 1, \\ (\mathsf{View}^{(i)}_3, k^{(i)}_2, k^{(i)}_3, x^{(i)}_3) & \text{if } e^{(i)} = 2, \\ (\mathsf{View}^{(i)}_1, k^{(i)}_3, k^{(i)}_1, x^{(i)}_3) & \text{if } e^{(i)} = 3. \end{cases}$$

4. Output $p \leftarrow [e, (b^{(1)}, z^{(1)}), (b^{(2)}, z^{(2)}), \cdots, (b^{(t)}, z^{(t)})]$.

Scheme 5: The zero knowledge simulator

which make q_0, q_1 queries to H respectively and classical A_C, all three of which may share state, let P_C be the probability if we choose a random function H and a random output B, then run A_0^H followed by A_C to generate x, and then run $A_1^H(x,B)$, that for a random j, the j-th query A_1^H makes is measured as $x' = x$. Then as long as the output of A_C has collision-entropy at least k, the advantage with which A_1^H, when run after A_0, A_C as described, distinguishes (x,B) from $(x,H(x))$ is at most $(4+\sqrt{2})\sqrt{q_0}2^{-k/4} + 2q_1\sqrt{P_C}$. In other words, if we can divide our game into three such algorithms and argue that the A_1 queries H on something that collapses to x with only negligible probability, then we can conclude that the two games are indistinguishable. Let A_0 run the game up until just before the i th iteration in the proof generation. Let A_C be the process which chooses $k^{(i)}_1, k^{(i)}_2, k^{(i)}_3$ and generates $\mathsf{View}^{(i)}_1, \mathsf{View}^{(i)}_2, \mathsf{View}^{(i)}_3$, and outputs $x = k_{e*^{(i)}}, \mathsf{View}_{e*^{(i)}}$. (Note that this has collision entropy $|k_{e*^{(i)}}|$ which is sufficient.) Let A_1 be the process which runs the rest of the proof, and then runs the adversary on the response.

Now we just have to argue that the probability that we make a measurement of A_1's j-th query to H and get x is negligible. To do this, we reduce to the security of the PRG used to generate the random tapes (and hence the views). Note that besides the one RO query, $k_{e*^{(i)}}$ is only used as input to the PRG. So, suppose there exists a quantum adversary A for which the resulting

A_1 has non-negligible probability of making an H-query that collapses to x. Then we can construct a quantum attacker for the PRG: we run the above A_0, A_C, but instead of choosing $k_{e*^{(i)}}$ we use the PRG challenge as the resulting random tape, and return a random value as the RO output. Then we run A_1, which continues the proof (which should query $k_{e*^{(i)}}$ only with negligible probability since ks are chosen at random), and then runs the adversary. We pick a random j, and on the adversary's j-th query, we make a measurement and if it gives us a seed consistent with our challenge tape, we output 1, otherwise a random bit. If P_C is non-negligible then we will obtain the correct seed and distinguish with non-negligible probability.

Game 4: For each i instead of choosing random $k_{e*^{(i)}}$ and expanding it via the PRG to get the random tape used to compute the views, we choose those tapes directly at random.

Note that in Game 3, $k_{e*^{(i)}}$ are now only used as seeds for the PRG, so this follow from pseudorandomness via a hybrid argument.

Game 5: We use the simulator to generate the views that will be opened, i.e. $j \neq e*^{(i)}$ for each i. We note that now the simulator no longer uses the witness.

This is identical by perfect privacy of the circuit decomposition.

Game 6: To allow for extraction in the simulation-extractability game we replace the random oracles with random polynomials whose degree is larger than the number of queries the adversary makes. The argument here identical to that in [84].

Online Extractability Before we prove online simulation-extractability, we define some notation to simplify the presentation:

For any proof $\pi = e, \{b^{(i)}, g^{(i)}, z^{(i)}\}_{i=1\dots t}$, let $\mathsf{hash\text{-}input}(\pi) = \{a^{(i)}, h^{(i)} = (g_1^{(i)}, g_2^{(i)}, g_3^{(i)})\}$ be the values that the verifier uses as input to H_{chal} in the verification of π as described in Figure 1.

For a proof $\pi = (e, \{b^{(i)}, g^{(i)}, z^{(i)}\}_{i=1\dots t})$, let $\mathsf{open}_0(z^{(i)}), \mathsf{open}_1(z^{(i)})$ denote the values derived from $z^{(i)}$ and used to compute $C_{e_i}^{(i)}$ and $C_{e_i+1}^{(i)}$ respectively in the description of Ver in Figure 1.

We say a tuple $(a, j, (o_1, o_2))$ is valid if $a = (y_1, y_2, y_3, C_1, C_2, C_3)$, $C_j = H_{com}(o_1)$, $C_{j+1} = H_{com}(o_2)$ and o_1, o_2 consist of k, View pairs for player $j, j+1$ that are consistent according to the circuit decomposition. We say $(a, j, (O_1, O_2))$ is set-valid if there exists $o_1 \in O_1$ and $o_2 \in O_2$ such that $(a, j, (o_1, o_2))$ is valid and set-invalid if not.

We first restate lemma 16 from [84] tailored to our application, in particular the fact that our proofs do not explicitly contain the commitment but rather the information the verifier needs to recompute it.

Lemma 1. *Let q_G be the number of queries to G made by the adversary A and the simulator S in the simulation extractability game, and let n be the number of proofs generated by S. Then the probability that A produces $x, \pi^* \notin$ simproofs where x, π^* is accepted by Ver^H, and $\mathsf{hash\text{-}input}(\pi^*) =$ $\mathsf{hash\text{-}input}(\pi')$ for a previous proof π' produced by the simulator, is at most $n(n+1)/2(2^{-k})^{3t} + O((q_G+1)^3 2^{-k})$ (Call this event $\mathsf{MallSim}$.)*

Proof. This proof follows almost exactly as in [84].

First, we argue that G is indistinguishable from a random function exactly in [84].

Then, observe that there are only two ways $\mathsf{MallSim}$ can occur:

Let e' be the hash value in π'. Then either S reprograms H sometime after π' is generated so that $H(\mathsf{hash\text{-}input}(\pi'))$ is no longer e', or π^* also contains the same e as π, i.e. $e = e'$. S only reprograms H if it chooses the same $\mathsf{hash\text{-}input}$ in a later proof - and $\mathsf{hash\text{-}input}$ includes $g_j^{(i)}$, i.e. a random function applied to an input which includes a randomly chosen seed. Thus, the probability that S chooses the same $\mathsf{hash\text{-}input}$ twice is at most $n(n+1)/2(2^{-k})^{3t} + O((q_G+1)^3 2^{-k}$, where $(2^{-k})^{3t}$ is the probability that two proofs use all the same seeds, and $O((q_G+1)^3 2^{-k}$ is the probability that two different seeds result in a collision in G, where the latter follows from Theorem 8 in [84].

The other possibility is that $\mathsf{hash\text{-}input}(\pi^*) = \mathsf{hash\text{-}input}(\pi')$, and $e = e'$, but $b^{(i)}, g^{(i)}, z^{(i)} \neq b'^{(i)}, g'^{(i)}, z'^{(i)}$ for some i. First note, that if $e = e'$ and $\mathsf{hash\text{-}input}(\pi^*) = \mathsf{hash\text{-}input}(\pi')$, then $g^{(i)} = g'^{(i)}$ and $b^{(i)} = b'^{(i)}$ for all i, by definition of $\mathsf{hash\text{-}input}$. Thus, the only remaining possibility is that $z^{(i)} \neq z'^{(i)}$ for some i. But since $h^{(i)} = h'^{(i)}$ for all i, this implies a collision in G, which again by Theorem 8 in [84] occurs with probability at most $O((q_G+1)^3 2^{-k}$.

We conclude that $\mathsf{MallSim}$ occurs with probability at most $n(n+1)/2(2^{-k})^{3t} + O((q_G+1)^3 2^{-k}$. □

Here, next we present our variant of lemma 17 from [84]. Note that this is quoted almost directly from Unruh with two modifications to account for the fact that our proofs do not explicitly contain the commitment but rather the information the verifier needs to recompute it, and the fact that our underlying Σ-protocol has only 3 challenges and satisfies 3-special soundness. H_0 in this lemma will correspond in our final proof to the initial state of H_{chal}.

Lemma 2. *Let G, H_{com} be arbitrarily distributed functions, and let $H_0 : \{0,1\}^{\leq \ell} \to \{0,1\}^{2t}$ be uniformly random (and independent of G). , Then, it is hard to find x and π such that for $\{a^{(i)}, (g_1^{(i)}, g_2^{(i)}, g_3^{(i)})\} = \mathsf{hash\text{-}input}(\pi)$ and $J_1 || \ldots || J_t := H_0(\mathsf{hash\text{-}input}(\pi))$*

(i) $g_{J_i}^{(i)} = G(\mathsf{open}_0(z^{(i)}))$ *and* $g_{J_i+1}^{(i)} = G(\mathsf{open}_1(z^{(i)}))$ *for all i.*
(ii) $(a^{(i)}, J_i, (\mathsf{open}_0(z^{(i)}), \mathsf{open}_1(z^{(i)})))$ *is valid for all i.*
(iii) For every i, there exists a j such that $(a^{(i)}, j, G^{-1}(g_{i,j}), G^{-1}(g_{i,j+1})))$ is set-invalid.

More precisely, if A^{G,H_0} makes at most q_H queries to H_0, it outputs (x, π) with these properties with probability at most $2(q_H+1)(\frac{2}{3})^{t/2}$

Proof. Without loss of generality, we can assume that G, H_{com} are fixed functions which A knows, so for this lemma we only treat H_0 as a random oracle.

For any given value of H_0, we call a tuple $c = (x, \{a^{(i)}\}_i, \{g_j^{(i)}\}_{i,j})$ a candidate iff: for each i, among the three transcripts, $(a^{(i)}, 1, G^{-1}(g_1)^{(i)}, G^{-1}(g_2^{(i)}))$, $(a^{(i)}, 2, G^{-1}(g_2^{(i)}),$

$G^{-1}(g_3^{(i)})$), and $(a^{(i)}, 3, G^{-1}(g_3^{(i)}), G^{-1}(g_1^{(i)}))$ at least one is set-valid, and at least one is set-invalid. Let $n_{\mathsf{twovalid}}(c)$ be the number of i's for which there are 2 set-valid transcripts. Let $\mathsf{E}_{\mathsf{valid}}(c)$ be the set of challenge tuples which correspond to only set-valid conversations. (Note that $|\mathsf{E}_{\mathsf{valid}}(c)| = 2^{n_{\mathsf{twovalid}}(c)}$.) We call a candidate an H_0*-solution* if the challenge produced by H_0 only opens set-valid conversations, i.e. in lies in $\mathsf{E}_{\mathsf{valid}}(c)$. We now aim to prove that A^H outputs an H_0 solution with negligible probability.

For any given candidate c, for uniformly random H_0, the probability that c is an H_0*-solution* is $\leq (\frac{2}{3})^t$. In particular, for candidate c the probability is $(\frac{2}{3})^t * 2^{n_{\mathsf{twovalid}}(c)-t}$.

Let Cand be the set of all candidates. Let $F : \mathsf{Cand} \to \{0,1\}$ be a random function such that for each c $F(c)$ is i.i.d. with $Pr[F_1(c) = 1] = (2/3)^t$.

Given F, we construct $H_F : \{0,1\}^* \to \mathbb{Z}_3^t$ as follows:

- For each $c \notin \mathsf{Cand}$, $H_F(c)$ is set to a uniformly random $y \in \mathbb{Z}_3^t$.
- For each $c \in \mathsf{Cand}$ such that $F(c) = 0$, $H_F(c)$ is set to a uniformly random $y \in \mathbb{Z}_3^t \setminus \mathsf{E}_{\mathsf{valid}}(c)$.
- For each $c \in \mathsf{Cand}$ with $F(c) = 1$, with probability $2^{n_{\mathsf{twovalid}}-t}$, choose a random challenge tuple e from $\mathsf{E}_{\mathsf{valid}}(c)$, and set $H_F(c) := e$. Otherwise $H_F(c)$ is set to a uniformly random $y \in \mathbb{Z}_3^t \setminus \mathsf{E}_{\mathsf{valid}}(c)$.

Note that for each c, and e the probability of $H(c)$ being set to e is 3^{-t}. Suppose A_0^H outputs an H_0-solution with probability μ, then since H_F has the same distribution as H_0, $A^{H_F}()$ outputs an H_F solution c with probability μ. By our definition of H_F, if c is an H_F solution, then $F(c) = 1$. Thus, $A^{H_F}()$ outputs c such that $F(c) = 1$ with probability at least μ.

As in [84], we can simulate $A^{H_F}()$ with another algorithm which generates H_F on the fly, and thus makes at most the same number of queries to F that A makes to H_F. Thus by applying Lemma 7 from [84], we get

$$\mu \leq 2(q_H + 1)(\frac{2}{3})^{t/2}.$$

□

Finally, as the sigma protocol underlying our proofs is only computationally sound (because we use H_{com} for our commitment scheme), we need to argue that an extractor can extract from 3 valid transcripts with all but negligible probability.

Lemma 3. *There exists an extractor E_Σ such that for any ppt quantum adversary A, the probability that A can produce $(a, \{(\nu_{1,j}, \nu_{2,j})\}_{j=1,2,3})$ such that $(a, j, (\nu_{1,j}, \nu_{2,j}))$ is a valid transcript for $j = 1, 2, 3$, but $E_\Sigma(a, \{(\nu_{1,j}, \nu_{2,j})\}_{j=1,2,3})$ fails to extract a proof, is negligible.*

Proof. Recall that $a = (y_1, y_2, y_3, C_1, C_2, C_3)$, and if all three transcripts are valid, $C_j = H_{com}(\nu_{1,j}) = H_{com}(\nu_{2,j-1})$ for $j = 1, 2, 3$. Thus, either we have $\nu_{1,j} = \nu_{2,j-1}$ for all j or $\mathcal{A}$ has found a collision in H_{com}. But, Theorem 8 in [84] tells us that the probability of finding a collision in a random function with k-bit output using at most q queries is at most $O((q+1)^3 2^{-k})$, which is negligible. If $\nu_{1,j} = \nu_{2,j-1}$ for all j, then we have 3 $k_j||\mathsf{View}_j$ values, all of which are pairwise consistent, so we conclude by the correctness of the circuit decomposition, and the fact that $(x = y, w) \in R$ iff $\phi(w) = y$ that if we sum the input share in $\mathsf{View}_1, \mathsf{View}_2, \mathsf{View}_3$, we get a witness such that $(x, w) \in R$. □

Theorem 2. *Our version of the Unruh protocol satisfies simulation-extractability against a quantum adversary.*

Proof. We define the following extractor:

1. On input π, compute $\mathsf{hash\text{-}input}(\pi) = \{a^{(i)}, h^{(i)} = (g_1^{(i)}, g_2^{(i)}, g_3^{(i)})\}$
2. For $i \in 1, \dots, t$: For $j \in 1, 2, 3$, check whether there is a solution $\nu_{1,j} \in G^{-1}(g_j^{(i)}), \nu_{2,j} \in G^{-1}(g_{j+1}^{(i)})$ such that $(a^{(i)}, j, (\nu_{1,j}, \nu_{2,j}))$ is a valid transcript. If there is a valid transcript for all j, output $E_\Sigma(a^{(i)}, \{(\nu_{1,j}, \nu_{2,j})\}_{j=1,2,3})$ as defined by Lemma 3 and halt.
3. If no solution is found, output $\perp$.

First we define some notation, again borrowed heavily from [84]:

Let $\mathsf{Ev}_i, \mathsf{Ev}_{ii}, \mathsf{Ev}_{iii}$ be events denoting that A in the simulation extractability game produces a proof satisfying conditions *(i)*, *(ii)*, and *(iii)* from Lemma 2 respectively.

Let $\mathsf{SigExtFail}$ be the event that the extractor finds a successful $(a, \{(\nu_{1,j}, \nu_{2,j})\}_{j=1,2,3})$, but E_Σ fails to produce a valid witness.

Let $\mathsf{ShouldExt}$ denote the event that A produces x, π such that Ver^H accepts and $(x, \pi) \notin$ *simproofs.*

Then our goal is to prove that the w produced by the extractor is such that $(x, w) \in R$. I.e., we want to prove that the following probability is negligible.

$$\begin{aligned}
&\Pr[\mathsf{ShouldExt} \wedge (x,w) \notin R] \\
\leq\, &\Pr[\mathsf{ShouldExt} \wedge (x,w) \notin R \wedge \neg\mathsf{MallSim}] \\
&+ \Pr[\mathsf{MallSim}] \\
=\, &\Pr[\mathsf{ShouldExt} \wedge (x,w) \notin R \wedge \neg\mathsf{MallSim} \wedge \neg\mathsf{Ev}_{iii}] \\
&+ \Pr[\mathsf{ShouldExt} \wedge (x,w) \notin R \wedge \neg\mathsf{MallSim} \wedge \mathsf{Ev}_{iii}] \\
&+ \Pr[\mathsf{MallSim}] \\
\leq\, &\Pr[(x,w) \notin R \wedge \neg\mathsf{Ev}_{iii}] \\
&+ \Pr[\mathsf{ShouldExt} \wedge (x,w) \notin R \wedge \neg\mathsf{MallSim} \wedge \mathsf{Ev}_{iii}] \\
&+ \Pr[\mathsf{MallSim}] \\
=\, &\Pr[\mathsf{SigExtFail}] \\
&+ \Pr[\mathsf{ShouldExt} \wedge (x,w) \notin R \wedge \neg\mathsf{MallSim} \wedge \mathsf{Ev}_{iii}] \\
&+ \Pr[\mathsf{MallSim}] \\
=\, &\Pr[\mathsf{SigExtFail}] \\
&+ \Pr[\mathsf{ShouldExt} \wedge (x,w) \notin R \wedge \neg\mathsf{MallSim} \wedge \mathsf{Ev}_{i} \wedge \mathsf{Ev}_{ii} \wedge \mathsf{Ev}_{iii}] \\
&+ \Pr[\mathsf{MallSim}] \\
\leq\, &\Pr[\mathsf{SigExtFail}] \\
&+ \Pr[\mathsf{Ev}_{i} \wedge \mathsf{Ev}_{ii} \wedge \mathsf{Ev}_{iii}] \\
&+ \Pr[\mathsf{MallSim}]
\end{aligned}$$

Here, the second equality follows from the definition of SigExtFail and Ev_{iii}, and the description of the extractor. The third equality follows from the fact that ¬MallSim means that the hash function on hash-input(π) has not been reprogrammed, and the fact that ShouldExt means verification succeeds, which means that conditions *(i)* and *(ii)* are satisfied.

Finally, by Lemmas 3, 2, and 1, we conclude that this probability is negligible. □

6
Post-Quantum Zero-Knowledge Proofs for Accumulators with Applications to Ring Signatures from Symmetric-Key Primitives

Publication Data

David Derler, Sebastian Ramacher, and Daniel Slamanig. Post-Quantum Zero-Knowledge Proofs for Accumulators with Applications to Ring Signatures from Symmetric-Key Primitives. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings*, volume 10786 of *Lecture Notes in Computer Science*, pages 419–440. Springer, 2018. URL: https://doi.org/10.1007/978-3-319-79063-3_20

The appended paper is an author-created extended version available at https://eprint.iacr.org/2017/1154. The extended version discusses further improvements to the signature size.

Contributions

- The author is one of the main authors.

Post-Quantum Zero-Knowledge Proofs for Accumulators with Applications to Ring Signatures from Symmetric-Key Primitives[⋆]

David Derler[1], Sebastian Ramacher[1], and Daniel Slamanig[2]

[1] IAIK, Graz University of Technology, Graz, Austria
[2] AIT Austrian Institute of Technology, Vienna, Austria
firstname.lastname@tugraz.at, firstname.lastname@ait.ac.at

Abstract. In this paper we address the construction of privacy-friendly cryptographic primitives for the post-quantum era and in particular accumulators with zero-knowledge membership proofs and ring signatures. This is an important topic as it helps to protect the privacy of users in online authentication or emerging technologies such as cryptocurrencies. Recently, we have seen first such constructions, mostly based on assumptions related to codes and lattices. We, however, ask whether it is possible to construct such primitives without relying on structured hardness assumptions, but solely based on symmetric-key primitives such as hash functions or block ciphers. This is interesting because the resistance of latter primitives to quantum attacks is quite well understood.

In doing so, we choose a modular approach and firstly construct an accumulator (with one-way domain) that allows to efficiently prove knowledge of (a pre-image of) an accumulated value in zero-knowledge. We, thereby, take care that our construction can be instantiated solely from symmetric-key primitives and that our proofs are of sublinear size. Latter is non trivial to achieve in the symmetric setting due to the absence of algebraic structures which are typically used in other settings to make these efficiency gains. Regarding efficient instantiations of our proof system, we rely on recent results for constructing efficient non-interactive zero-knowledge proofs for general circuits. Based on this building block, we then show how to construct logarithmic size ring signatures solely from symmetric-key primitives. As constructing more advanced primitives only from symmetric-key primitives is a very recent field, we discuss some interesting open problems and future research directions. Finally, we want to stress that our work also indirectly impacts other fields: for the first time it raises the requirement for collision resistant hash functions with particularly low AND count.

Keywords: post-quantum cryptography, privacy-preserving cryptography, provable security, accumulator, zero-knowledge for circuits

[⋆] This is the extended version of a paper which appears in Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings. ©Springer, 2018. The final publication is available at Springer via https://doi.org/10.1007/978-3-319-79063-3_20.

1 Introduction

The design of cryptographic schemes that remain secure in the advent of powerful quantum computers has become an important topic in recent years. Although it is hard to predict when quantum computers will be powerful enough to break factoring and discrete logarithm based cryptosystems, it is important to start the transition to post-quantum cryptography early enough to eventually not end up in a rush. This is underpinned by the NIST post-quantum cryptography standardization project[3], which aims at identifying the next generation of public key encryption, key exchange and digital signature schemes basing their security on conjectured quantum hard problems. Apart from these fundamental schemes, there are many other valuable schemes which would nicely complement a post-quantum cryptographic toolbox. In this paper we are interested in privacy-friendly cryptographic primitives for the post-quantum era and in particular accumulators with zero-knowledge membership proofs and ring signatures. Such schemes help to protect the privacy of users, and significantly gained importance due to recent computing trends such as Cloud computing or the Internet of Things (IoT). Examples where privacy-enhancing protocols are already widely deployed today are remote attestation via direct anonymous attestation (DAA) [BCC04] as used by the Trusted Platform Module (TPM)[4], privacy-friendly online authentication within Intel's Enhanced Privacy ID (EPID) [BL07], or usage within emerging technologies such as cryptocurrencies to provide privacy of transactions.[5]

Let us now briefly discuss the primitives we construct in this paper. An accumulator scheme [BdM93] allows to represent a finite set as a succinct value called the accumulator. For every element in the accumulated set, one can efficiently compute a so called witness to certify its membership in the accumulator. However, it should be computationally infeasible to find a witness for non-accumulated values. We are interested in accumulators supporting efficient zero-knowledge membership proofs. Ring signature schemes [RST01] allow a member of an ad-hoc group $\mathcal{R}$ (the so called ring), defined by the member's public keys, to anonymously sign a message on behalf of $\mathcal{R}$. Such a signature attests that some member of $\mathcal{R}$ produced the signature, but the actual signer remains anonymous.

For ring signatures there is a known approach to construct them from accumulators and non-interactive zero-knowledge proof systems in the random oracle model. The main technical hurdle in the post-quantum setting is to find accumulators, and, more importantly, compatible proof systems under suitable assumptions. Only recently, Libert et al. in [LLNW16] showed that it is possible to instantiate this approach in the post-quantum setting and provided the first post-quantum accumulator from lattices. This combined with suitable non-interactive variants of Σ-protocols yields post-quantum ring signatures in the

[3] https://csrc.nist.gov/groups/ST/post-quantum-crypto/

[4] https://trustedcomputinggroup.org/tpm-library-specification/

[5] https://getmonero.org/resources/moneropedia/ringsignatures.html

random oracle model (ROM). However, this does not give rise to a construction of ring signatures from symmetric-key primitives such as hash functions or block ciphers, as we pursue in this paper. The main technical tools we use in our construction are recent results from zero-knowledge proof systems for general circuits [GMO16, CDG^{+}17], and our techniques are inspired by recent approaches to construct post-quantum signature schemes based on these proof systems [CDG^{+}17]. We note that there are also post-quantum ring signature candidates from problems related to codes [MCG08] and multivariate cryptography [MP17]. However, they all have size linear in the number of ring members, whereas we are only interested in sublinear ones. Additionally, former schemes are proven secure in weaker security models.

Contribution. Our contributions can be subsumed as follows:

- We present the first post-quantum accumulator (with one-way domain) together with efficient zero-knowledge proofs of (a pre-image of) an accumulated value, which solely relies on assumptions related to symmetric-key primitives. That is, we do not require any structured hardness assumptions. Our proofs are of sublinear size in the number of accumulated elements and can be instantiated in both, the ROM as well as the quantum accessible ROM (QROM). Besides being used as an important building block in this paper, such accumulators are of broader interest. In particular, such accumulators with efficient zero-knowledge membership proofs have many other applications beyond this work, e.g., membership revocation [BCD^{+}17] or anonymous cash such as Zerocoin [MGGR13]. We also note that the only previous construction of post-quantum accumulators with efficient zero-knowledge membership proofs in [LLNW16] relies on hardness assumptions on lattices.
- We use our proposed accumulator to construct ring signatures of sublinear size. Therefore, we prove an additional property—simulation-sound extractability—of the proof system (ZKB++ [CDG^{+}17]) we are using. This then allows us to rigorously prove the security of our ring signature construction in the strongest model of security for ring signatures due to Bender et al. [BKM09]. Consequently, we propose a construction of sublinear size ring signatures solely from symmetric-key primitives.
- We present a selection of symmetric-key primitives that can be used to instantiate our ring signature construction and evaluate the practicality of our approach. In particular, we present signature sizes for rings of various sizes when instantiating the one-way function and hash function using LowMC [ARS^{+}15, ARS^{+}16]. Finally, we present some interesting directions for future research within this very recent domain.

Additional Contribution Compared to PQCrypto'18 Version. We propose a concrete, optimized implementation of the circuit used in the zero-knowledge membership proof for the accumulator. Our techniques roughly allow reduce the proof (signature) sizes by a factor of 2 when compared to the circuit used for the evaluation in the PQCrypto'18 version. A recent work of Boneh et al. [BEF18], who construct post-quantum group signatures, also presents results allowing to optimize our zero-knowledge membership proof sizes compared to the PQCrypto'18 version. We want to emphasize that the results presented in this extended full paper allow for even smaller zero-knowledge membership proofs than what is obtained with the optimizations due to Boneh et al. in [BEF18]. Note that the optimizations presented in this version also allows to instantiate the group signature scheme [BEF18, Construction II] with smaller signature sizes.

2 Preliminaries

Notation. Let $x \xleftarrow{R} X$ denote the operation that picks an element uniformly at random from a finite set X and assigns it to x. We assume that all algorithms run in polynomial time and use $y \leftarrow \mathsf{A}(x)$ to denote that y is assigned the output of the potentially probabilistic algorithm A on input x and fresh random coins. For algorithms representing adversaries we use calligraphic letters, e.g., $\mathcal{A}$. We assume that every algorithm outputs a special symbol $\perp$ on error. We write $\Pr[\Omega : \mathcal{E}]$ to denote the probability of an event $\mathcal{E}$ over the probability space Ω. A function $\epsilon : \mathbb{N} \to \mathbb{R}^+$ is called negligible if for all $c > 0$ there is a k_0 such that $\epsilon(k) < 1/k^c$ for all $k > k_0$. In the remainder of this paper, we use ϵ to denote such a negligible function. Finally, we define $[n] \coloneqq \{1, \ldots, n\}$.

2.1 Zero-Knowledge Proofs and Σ-Protocols

Σ-Protocols. Let $L \subseteq \mathsf{X}$ be an **NP**-language with witness relation R so that $L = \{x \mid \exists w : R(x, w) = 1\}$. A Σ-protocol for language L is defined as follows.

Definition 1 (Σ-Protocol). *A Σ-protocol for language L is an interactive three-move protocol between a PPT prover $\mathsf{P} = (\mathsf{Commit}, \mathsf{Prove})$ and a PPT verifier $\mathsf{V} = (\mathsf{Challenge}, \mathsf{Verify})$, where P makes the first move and transcripts are of the form $(\mathsf{a}, \mathsf{e}, \mathsf{z}) \in \mathsf{A} \times \mathsf{E} \times \mathsf{Z}$, where a is output by Commit, e is output by $\mathsf{Challenge}$ and z is output by Prove. Additionally, Σ protocols satisfy the following properties*

Completeness. *For all security parameters κ, and for all $(x, w) \in R$, it holds that*

$$\Pr[\langle \mathsf{P}(1^\kappa, x, w), \mathsf{V}(1^\kappa, x)\rangle = 1] = 1.$$

s-Special Soundness. *There exists a PPT extractor E so that for all x, and for all sets of accepting transcripts $\{(\mathsf{a}, \mathsf{e}_i, \mathsf{z}_i)\}_{i \in [s]}$ with respect to x where $\forall i, j \in [s], i \neq j : \mathsf{e}_i \neq \mathsf{e}_j$, generated by any algorithm with polynomial runtime in κ, it holds that*

$$\Pr\left[w \leftarrow \mathsf{E}(1^\kappa, x, \{(\mathsf{a}, \mathsf{e}_i, \mathsf{z}_i)\}_{i \in [s]}) : (x, w) \in R\right] \geq 1 - \epsilon(\kappa).$$

Special Honest-Verifier Zero-Knowledge. *There exists a PPT simulator S so that for every $x \in L$ and every challenge $\mathsf{e} \in \mathsf{E}$, it holds that a transcript $(\mathsf{a}, \mathsf{e}, \mathsf{z})$, where $(\mathsf{a}, \mathsf{z}) \leftarrow \mathsf{S}(1^\kappa, x, \mathsf{e})$ is computationally indistinguishable from a transcript resulting from an honest execution of the protocol.*

The s-special soundness property gives an immediate bound for soundness: if no witness exists then (ignoring a negligible error) the prover can successfully answer at most to $(s-1)/t$ challenges, where $t = |\mathsf{E}|$ is the size of the challenge space. In case this value is too large, it is possible to reduce the soundness error using ℓ-fold parallel repetition of the Σ-protocol. Furthermore, it is also well known that one can easily express conjunctions and disjunctions of languages proven using Σ-protocols. For the formal details refer to [Dam10, CDS94].

Non-Interactive ZK Proof Systems. Now, we recall a standard definition of non-interactive zero-knowledge proof systems. Therefore, let L be an **NP**-language with witness relation R so that $L = \{x \mid \exists\ w : R(x, w) = 1\}$.

Definition 2 (Non-Interactive Zero-Knowledge Proof System). *A non-interactive proof system* Π *is a tuple of algorithms* (Setup, Proof, Verify)*, defined as:*

$\mathsf{Setup}(1^\kappa)$: *This algorithm takes a security parameter κ as input, and outputs a common reference string* crs.

$\mathsf{Proof}(\mathsf{crs}, x, w)$: *This algorithm takes a common reference string* crs*, a statement x, and a witness w as input, and outputs a proof π.*

$\mathsf{Verify}(\mathsf{crs}, x, \pi)$: *This algorithm takes a common reference string* crs*, a statement x, and a proof π as input, and outputs a bit $b \in \{0, 1\}$.*

We require the properties *completeness*, *adaptive zero-knowledge*, and *simulation-sound extractability* as defined below.

Definition 3 (Completeness). *A non-interactive proof system* Π *is* complete*, if for every adversary $\mathcal{A}$ it holds that*

$$\Pr\left[\begin{matrix}\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa),\ (x, w) \leftarrow \mathcal{A}(\mathsf{crs}), \\ \pi \leftarrow \mathsf{Proof}(\mathsf{crs}, x, w)\end{matrix} \ :\ \begin{matrix}\mathsf{Verify}(\mathsf{crs}, x, \pi) = 1 \\ \vee\ (x, w) \notin R\end{matrix}\right] \approx 1.$$

Definition 4 (Adaptive Zero-Knowledge). *A non-interactive proof system* Π *is* adaptively zero-knowledge*, if there exists a PPT simulator $S = (\mathcal{S}_1, \mathcal{S}_2)$ such that for every PPT adversary $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that*

$$\left|\begin{matrix}\Pr\left[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa)\ :\ \mathcal{A}^{\mathcal{P}(\mathsf{crs},\cdot,\cdot)}(\mathsf{crs}) = 1\right]\ - \\ \Pr\left[(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^\kappa)\ :\ \mathcal{A}^{\mathcal{S}(\mathsf{crs},\tau,\cdot,\cdot)}(\mathsf{crs}) = 1\right]\end{matrix}\right| \leq \epsilon(\kappa),$$

where, τ denotes a simulation trapdoor. Thereby, $\mathcal{P}$ and $\mathcal{S}$ return $\perp$ if $(x, w) \notin R$ or $\pi \leftarrow \mathsf{Proof}(\mathsf{crs}, x, w)$ and $\pi \leftarrow \mathcal{S}_2(\mathsf{crs}, \tau, x)$, respectively, otherwise.

Definition 5 (Simulation-Sound Extractability). *An adaptively zero-knowledge non-interactive proof system* Π *is simulation-sound extractable, if there exists a PPT extractor $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$ such that for every adversary $\mathcal{A}$ it holds that*

$$\left|\begin{matrix}\Pr\left[(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^\kappa)\ :\ \mathcal{A}(\mathsf{crs}, \tau) = 1\right]\ - \\ \Pr\left[(\mathsf{crs}, \tau, \xi) \leftarrow \mathcal{E}_1(1^\kappa)\ :\ \mathcal{A}(\mathsf{crs}, \tau) = 1\right]\end{matrix}\right| = 0,$$

and for every PPT adversary $\mathcal{A}$ there is a negligible function $\varepsilon_2(\cdot)$ such that

$$\Pr\left[\begin{array}{l}(\mathsf{crs},\tau,\xi)\leftarrow\mathcal{E}_1(1^\kappa),\\(x^\star,\pi^\star)\leftarrow\mathcal{A}^{\mathcal{S}(\mathsf{crs},\tau,\cdot)}(\mathsf{crs}),\\w\leftarrow\mathcal{E}_2(\mathsf{crs},\xi,x^\star,\pi^\star)\end{array}\ :\ \begin{array}{r}\mathsf{Verify}(\mathsf{crs},x^\star,\pi^\star)=1\ \wedge\\(x^\star,\pi^\star)\notin\mathcal{Q}_\mathsf{S}\ \wedge\ (x^\star,w)\notin R\end{array}\right]\leq\varepsilon_2(\kappa),$$

where $\mathcal{S}(\mathsf{crs},\tau,x) := \mathcal{S}_2(\mathsf{crs},\tau,x)$ and $\mathcal{Q}_\mathsf{S}$ keeps track of the queries to and answers of $\mathcal{S}$.

The Fiat-Shamir Transform. The Fiat-Shamir transform [FS86] is a frequently used tool to convert Σ-protocols $\langle\mathsf{P},\mathsf{V}\rangle$ to their non-interactive counterparts. Essentially, the transform removes the interaction between P and V by using a RO $H : \mathsf{A}\times\mathsf{X}\to\mathsf{E}$ to obtain the challenge e.[6] That is, one uses a PPT algorithm $\mathsf{Challenge}'(1^\kappa,\mathsf{a},x)$ which obtains $\mathsf{e}\leftarrow H(\mathsf{a},x)$ and returns e. Then, the prover can locally obtain the challenge e *after* computing the initial message a. Starting a verifier $\mathsf{V}' = (\mathsf{Challenge}',\mathsf{Verify})$ on the same initial message a will then yield the same challenge e. More formally, we obtain the non-interactive PPT algorithms $(\mathsf{P}_H,\mathsf{V}_H)$ indexed by the used RO:

$\mathsf{P}_H(1^\kappa,x,w)$: Start P on $(1^\kappa,x,w)$, obtain the first message a, answer with $\mathsf{e}\leftarrow H(\mathsf{a},x)$, and finally obtain z. Returns $\pi\leftarrow(\mathsf{a},\mathsf{z})$.

$\mathsf{V}_H(1^\kappa,x,\pi)$: Parse π as (a,z). Start V' on $(1^\kappa,x)$, send a as first message to V'. When V' outputs e, reply with z and output 1 if V' accepts and 0 otherwise.

One can obtain a non-interactive proof system satisfying the properties above by applying the Fiat-Shamir transform to any Σ-protocol where the min-entropy α of the commitment a sent in the first phase is so that $2^{-\alpha}$ is negligible in the security parameter κ and the challenge space E is exponentially large in the security parameter. Formally, $\mathsf{Setup}(1^\kappa)$ fixes a RO $H : \mathsf{A}\times\mathsf{X}\to\mathsf{E}$, sets $\mathsf{crs}\leftarrow(1^\kappa,H)$ and returns crs. The algorithms Proof and Verify are defined as follows: $\mathsf{Proof}(\mathsf{crs},x,w) := \mathsf{P}_H(1^\kappa,x,w)$, $\mathsf{Verify}(\mathsf{crs},x,\pi) := \mathsf{V}_H(1^\kappa,x,\pi)$.

Signatures via Fiat-Shamir. The Fiat-Shamir (FS) transform can elegantly be used to convert (canonical) identification schemes into adaptively secure signature schemes. The basic idea is similar to above, but slightly differs regarding the challenge generation, i.e., one additionally includes the message upon generating the challenge. Note that in the context of the stronger variant of the FS transform we rely on, one can simply modify the language so that the statements additionally include the message to be signed. This is because our variant of the FS transform includes the statement upon challenge generation, which is why extending the statement by the message also implicitly means including the message in the challenge generation. We will not make this language change explicit in the following, but implicitly assume that the language is changed if a message is included as the last parameter of the statement to be proven.

[6] This is a stronger variant of FS (cf. [FKMV12, BPW12]). The original weaker variant of the FS transform does not include the statement x in the challenge computation.

The Unruh Transform. Similar to FS, Unruh's transform [Unr12, Unr15, Unr16] allows one to construct NIZK proofs and signature schemes from Σ-protocols. In contrast to the FS transform, Unruh's transform can be proven secure in the QROM (quantum random oracle model), strengthening the security guarantee against quantum adversaries. At a high level, Unruh's transform works as follows: given Σ-protocol, the prover repeats the first phase of the Σ-protocol t times and for each of those runs produces responses for M randomly selected challenges. All those responses are permuted using a random permutation G. Querying the random oracle on all first rounds all permuted responses then determines the responses to publish for each round.

2.2 Efficient NIZK Proof Systems for General Circuits

ZKB++ [CDG+17], an optimized version of ZKBoo [GMO16], is a proof system for zero-knowledge proofs over arbitrary circuits. ZKBoo and ZKB++ build on the MPC-in-the-head paradigm by Ishai et al. [IKOS09], which roughly works as follows. The prover simulates all parties of a multiparty computation protocol (MPC) implementing the joint evaluation of some function, say $y = \text{SHA-256}(x)$, and computes commitments to the states of all players. The verifier then randomly corrupts a subset of the players and checks whether those players did the computation correctly.

ZKBoo generalizes the idea of [IKOS09] by replacing MPC with circuit decompositions. There the idea is to decompose the circuit into three shares, where revealing the wire values of two shares does not leak any information about the wire values on the input of the circuit. The explicit formulas for circuit decomposition can be found in [GMO16] for ZKBoo and in [CDG+17] for ZKB++. Multiplication gates induce some dependency between the individual shares which is why the wire values on the output of the multiplication gates needs to be stored in the transcripts. Hence, the transcripts grow linearly in the number of multiplication gates. Due to space limitations we do not include further details on ZKB++ and refer the reader to [CDG+17] for the details.

3 PQ Accumulators & ZK Membership Proofs

Our goal is to come up with an accumulator and associated efficient zero-knowledge membership proof system, which remains secure in the face of attacks by a quantum attacker. The first building block we, thus, require for our constructions are accumulators which can be proven secure under an assumption which is believed to resist attacks by a quantum computer. In this work our goal is to solely rely on unstructured assumptions, and thus resort to using Merkle tree as accumulators. Merkle trees were first used in the context of accumulators by Buldas, Laud, and Lipmaa in [BLL00], who called their primitive undeniable attesters. In the fashion of [DKNS04], we then extend the accumulator model to accumulators with one-way domain, i.e., accumulators where the accumulation domain coincides with the range of a one-way function so that one can accumulate images of the one-way function. For the associated zero-knowledge membership proof system, we build up on recent progress in proving statements over general circuits as discussed in Section 2.2.

The main technical hurdle we face in this context is designing the statement to be proven with the proof system so that we can actually obtain proofs which are sublinear (in particular logarithmic) in the number of accumulated elements. Obtaining sublinear proofs is complicated mainly due to the absence of any underlying algebraic structure on the accumulator.

3.1 Formal Model

We rely on the formalization of accumulators by [DHS15], which we slightly adapt to fit our requirement for a deterministic Eval algorithm. Based on this formalization we then restate the Merkle tree accumulator (having a deterministic Eval algorithm) within this framework.

Definition 6 (Accumulator). *A static accumulator is a tuple of efficient algorithms* (Gen, Eval, $\mathsf{WitCreate}$, Verify) *which are defined as follows:*

$\mathsf{Gen}(1^\kappa, t)$: *This algorithm takes a security parameter κ and a parameter t. If $t \neq \infty$, then t is an upper bound on the number of elements to be accumulated. It returns a key pair* $(\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda)$*, where $\mathsf{sk}_\Lambda = \emptyset$ if no trapdoor exists. We assume that the accumulator public key pk_Λ implicitly defines the accumulation domain D_Λ.*

$\mathsf{Eval}((\mathsf{sk}_\Lambda^\sim, \mathsf{pk}_\Lambda), \mathcal{X})$: *This deterministic algorithm takes a key pair* $(\mathsf{sk}_\Lambda^\sim, \mathsf{pk}_\Lambda)$ *and a set $\mathcal{X}$ to be accumulated and returns an accumulator $\Lambda_\mathcal{X}$ together with some auxiliary information* aux.

$\mathsf{WitCreate}((\mathsf{sk}_\Lambda^\sim, \mathsf{pk}_\Lambda), \Lambda_\mathcal{X}, \mathsf{aux}, x_i)$: *This algorithm takes a key pair* $(\mathsf{sk}_\Lambda^\sim, \mathsf{pk}_\Lambda)$*, an accumulator $\Lambda_\mathcal{X}$, auxiliary information* aux *and a value x_i. It returns $\perp$, if $x_i \notin \mathcal{X}$, and a witness wit_{x_i} for x_i otherwise.*

$\mathsf{Verify}(\mathsf{pk}_\Lambda, \Lambda_\mathcal{X}, \mathsf{wit}_{x_i}, x_i)$: *This algorithm takes a public key pk_Λ, an accumulator $\Lambda_\mathcal{X}$, a witness wit_{x_i} and a value x_i. It returns 1 if wit_{x_i} is a witness for $x_i \in \mathcal{X}$ and 0 otherwise.*

We require accumulators to be correct and collision free. While we omit the straight forward correctness notion, we recall the collision freeness notion below, which requires that finding a witness for a non-accumulated value is hard.

Definition 7 (Collision Freeness). *A cryptographic accumulator is* collision-free*, if for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that:*

$$\Pr\left[\begin{array}{l}(\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda) \leftarrow \mathsf{Gen}(1^\kappa, t),\\ (\mathsf{wit}^\star_{x_i}, x^\star_i, \mathcal{X}^\star) \leftarrow \mathcal{A}(\mathsf{pk}_\Lambda)\end{array} : \begin{array}{r}\mathsf{Verify}(\mathsf{pk}_\Lambda, \Lambda^\star,\\ \mathsf{wit}^\star_{x_i}, x^\star_i) = 1 \wedge\\ x^\star_i \notin \mathcal{X}^\star\end{array}\right] \leq \varepsilon(\kappa),$$

where $\Lambda^\star \leftarrow \mathsf{Eval}((\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda), \mathcal{X}^\star)$.

$\underline{\mathsf{Gen}(1^\kappa, t)}$: Fix a family of hash functions $\{H_k\}_{k \in \mathsf{K}^\kappa}$ with $H_k : \{0,1\}^* \rightarrow \{0,1\}^\kappa \; \forall \, k \in \mathsf{K}^\kappa$. Choose $k \overset{R}{\leftarrow} \mathsf{K}^\kappa$ and return $(\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda) \leftarrow (\emptyset, H_k)$.

$\underline{\mathsf{Eval}((\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda), \mathcal{X})}$: Parse pk_Λ as H_k and $\mathcal{X}$ as $(x_0, \ldots, x_{n-1})$.

We assume without loss of generality that $\mathcal{X}$ is an ordered sequence instead of a set. If $\nexists \; k \in \mathbb{N}$ so that $n = 2^k$ return $\perp$. Otherwise, let $\ell_{u,v}$ refer to the u-th leaf (the leftmost leaf is indexed by 0) in the v-th layer (the root is indexed by 0) of a perfect binary tree. Return $\Lambda_\mathcal{X} \leftarrow \ell_{0,0}$ and $\mathsf{aux} \leftarrow ((\ell_{u,v})_{u \in [n/2^{k-v}]})_{v \in [k]}$, where

$$\ell_{u,v} \leftarrow \begin{cases} H_k(\ell_{2u,v+1} || \ell_{2u+1,v+1}) & \text{if } v < k \text{, and} \\ H_k(x_i) & \text{if } v = k. \end{cases}$$

$\underline{\mathsf{WitCreate}((\mathsf{sk}_{\tilde{\Lambda}}, \mathsf{pk}_\Lambda), \Lambda_\mathcal{X}, \mathsf{aux}, x_i)}$: Parse aux as $((\ell_{u,v})_{u \in [n/2^{k-v}]})_{v \in [k]}$ and return wit_{x_i} where

$$\mathsf{wit}_{x_i} \leftarrow (\ell_{\lfloor i/2^v \rfloor + \eta, k-v})_{0 \leq v \leq k}, \text{ where } \eta = \begin{cases} 1 & \text{if } \lfloor i/2^v \rfloor \pmod 2 = 0 \\ -1 & \text{otherwise.} \end{cases}$$

$\underline{\mathsf{Verify}(\mathsf{pk}_\Lambda, \Lambda_\mathcal{X}, \mathsf{wit}_{x_i}, x_i)}$: Parse pk_Λ as H_k, $\Lambda_\mathcal{X}$ as $\ell_{0,0}$, set $\ell_{i,k} \leftarrow H_k(x_i)$. Recursively check for all $0 < v < k$ whether the following holds and return 1 if so. Otherwise return 0.

$$\ell_{\lfloor i/2^{v+1} \rfloor, k-(v+1)} = \begin{cases} H_k(\ell_{\lfloor i/2^v \rfloor, k-v} || \ell_{\lfloor i/2^v \rfloor + 1, k-v}) & \text{if } \lfloor i/2^v \rfloor \pmod 2 = 0 \\ H_k(\ell_{\lfloor i/2^v \rfloor - 1, k-v} || \ell_{\lfloor i/2^v \rfloor, k-v}) & \text{otherwise.} \end{cases}$$

Scheme 1. Merkle tree accumulator.

3.2 The Accumulator

In Scheme 1, we cast the Merkle tree accumulator in the framework of [DHS15]. Then, we restate some well-known lemmas and sketch the respective proofs.

Lemma 1. *Scheme 1 is correct.*

The lemma above is easily verified by inspection. The proof is omitted.

Lemma 2. *If $\{H_k\}_{k \in \mathsf{K}^\kappa}$ is a family of collision resistant hash functions, the accumulator in Scheme 1 is collision free.*

Proof (Sketch). Upon setup, the reduction engages with a collision resistance challenger for the family of hash functions, obtains H_k, and completes the setup as in the original protocol. Now, one may observe that every collision in the accumulator output by the adversary implies that the reduction knows at least two colliding inputs for H_k, which upper bounds the probability of a collision in the accumulator by the collision probability of the hash function.

3.3 Accumulators with One-Way Domain

We now extend the definition of accumulators to ones with one-way domain following the definition of [DKNS04], but we adapt it to our notation.

Definition 8 (Accumulator with One-Way Domain). *A collision-free accumulator with accumulation domain* D_Λ *and associated function family* $\{f_\Lambda : \mathsf{I}_\Lambda \rightarrow \mathsf{D}_\Lambda\}$ *where* $\mathsf{Gen}(1^\kappa, t)$ *also selects* f_Λ *is called an accumulator with one-way domain if*

Efficient Verification. *There exists an efficient algorithm* D *that on input* $(x, z) \in \mathsf{D}_\Lambda \times \mathsf{I}_\Lambda$ *returns* 1 *if and only if* $f_\Lambda(z) = x$.

Efficient Sampling. *There exists a (probabilistic) algorithm* W *that on input* 1^κ *returns a pair* $(x, z) \in \mathsf{D}_\Lambda \times \mathsf{I}_\Lambda$ *with* $D(x, z) = 1$.

One-Wayness. *For all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\varepsilon(\cdot)$ *such that:*

$$\Pr\left[(x, z) \leftarrow W(1^\kappa), z^\star \leftarrow \mathcal{A}(1^\kappa, x) : D(x, z) = 1\right] \leq \varepsilon(\kappa).$$

Note that when we set f_Λ to be the identity function, then we have a conventional accumulator.

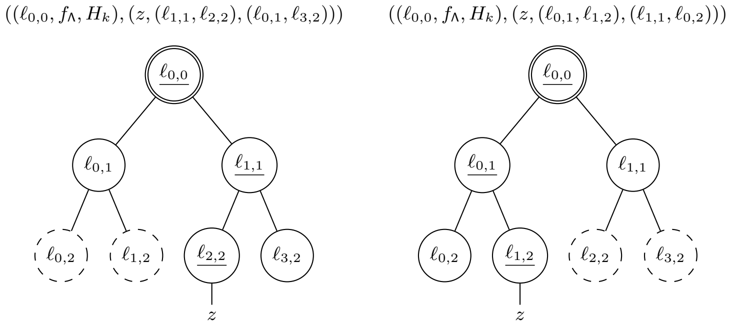
3.4 Membership Proofs of Logarithmic Size

The main technical tool used by [DKNS04] to obtain zero-knowledge membership proofs of constant size is to exploit a property of the accumulator which is called quasi-commutativity. Clearly, such a property requires some underlying algebraic structure which we explicitly want to sacrifice in favor of being able to solely rely on assumptions related to symmetric-key primitives with relatively well understood post-quantum security. To this end we have to use a different technique. First observe that when naïvely proving that a non-revealed value is a member of our accumulator would amount to a disjunctive proof of knowledge over all members, which is at least of linear size. Therefore, this is not an option and we have to develop an alternative technique.

The Relation. Essentially our idea is to "emulate" some kind of commutativity within the order of the inputs to the hash function in each level by a disjunctive proof statement, i.e., we exploit the disjunction to hide where the path through the tree continues. The single statements in every level of the tree are then included in one big conjunction. The length of this statement is $\mathcal{O}(k) = \mathcal{O}(\log n)$. More formally we define a relation R on $\{0,1\}^\kappa \times \{f_\Lambda\} \times \{H_k\} \times \mathsf{I}_\Lambda \times (\{0,1\}^\kappa)^{2k}$ which—for a given non-revealed pre-image z—attests membership of the corresponding image $f_\Lambda(z)$ in the accumulator $\Lambda_\mathcal{X}$:

$$\begin{aligned}((\Lambda_\mathcal{X}, f_\Lambda, H_k), (z, (a_i)_{i\in[k]}, (b_i)_{i\in[k]})) \in R \iff & (a_k = f_\Lambda(z) \ \vee \ b_k = f_\Lambda(z)) \\ & \wedge \ \bigwedge_{i=0}^{k-1} (a_i = H_k(a_{i+1}||b_{i+1}) \vee a_i = H_k(b_{i+1}||a_{i+1})),\end{aligned}$$

where $\Lambda_\mathcal{X} = a_0$. In Figure 1 we illustrate that the relation indeed works for arbitrary members of the accumulator without influencing the form of the statement or the witness. This illustrates that proving the statement in this way does not reveal any information on which path in the tree was taken. To see this, observe that at each level of the tree the relation covers both cases where a_i is either a left or right child. Given that, it is easy to verify that having a witness for relation R implies having a witness for the accumulator together with some (non-revealed) member.



$((\ell_{0,0}, f_\wedge, H_k), (z, (\ell_{1,1}, \ell_{2,2}), (\ell_{0,1}, \ell_{3,2})))$
$((\ell_{0,0}, f_\wedge, H_k), (z, (\ell_{0,1}, \ell_{1,2}), (\ell_{1,1}, \ell_{0,2})))$
$\ell_{0,0}$
$\ell_{0,1}$
$\ell_{1,1}$
$\ell_{0,2}$
$\ell_{1,2}$
$\ell_{2,2}$
$\ell_{3,2}$
z
$\ell_{0,0}$
$\ell_{0,1}$
$\ell_{1,1}$
$\ell_{0,2}$
$\ell_{1,2}$
$\ell_{2,2}$
$\ell_{3,2}$
z

Fig. 1. Visualization of different paths in the Merkle tree and the corresponding witness. The nodes on the path corresponding to a_0, a_1 and a_2 are underlined.

Remark 1. In order to use relation R with the conventional accumulator in Scheme 1, we just have to set $f_\wedge$ to be the identity function (which yields $x = z$) and then set $a_k = H_k(z)$ and $b_k = H_k(z)$.

3.5 Converting Accumulator Witnesses

Now, the remaining piece to finally be able to plug in a witness $\mathsf{wit}_{f_\wedge(z)}$ for some accumulated value $f_{\wedge(z)}$ with pre-image z into the relation R above is some efficient helper algorithm which rearranges the values z and $\mathsf{wit}_{f_\wedge(z)}$ so that they are compatible with the format required by R. Such an algorithm is easily implemented, which is why we only define the interface below.

$\mathsf{Trans}(z, \mathsf{wit}_{f_\wedge(z)})$: Takes as input a value z as well as a witness $\mathsf{wit}_{f_{\wedge(z)}}$ and returns a witness of the form $(z, (a_i)_{i\in[k]}, (b_i)_{i\in[k]})$ for R.

Since Trans can be viewed as a permutation on the indexes it is easy to see that the function implemented by Trans is bijective and its inverse is easy to compute. We denote the computation of the inverse of the function implemented by Trans as $(z, \mathsf{wit}_{f_\wedge(z)}) \leftarrow \mathsf{Trans}^{-1}(z, (a_i)_{i\in[n]}, (b_i)_{i\in[n]})$.

4 Logarithmic Size Ring Signatures

The two main lines of more recent work in the design of ring signatures target reducing the signature size or removing the requirement for random oracles (e.g., [DKNS04, CGS07, GK15, BCC$^+$15, DS16, Gon17, MS17]). We, however, note that all these approaches require assumptions that do not withstand a quantum computer. To the best of our knowledge, the first non-trivial post-quantum scheme (i.e., one that does not have linear size signatures) in the random oracle model is the lattice-based scheme recently proposed by Libert et al. [LLNW16]. We provide an alternative construction in the random oracle model with logarithmic sized signatures, but avoid lattice assumptions and only rely on symmetric-key primitives.

4.1 Formal Model

Below, we formally define ring signature schemes (adopting [BKM09]).

Definition 9 (Ring Signature). *A ring signature scheme* RS *is a tuple* $\mathsf{RS} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *of PPT algorithms, which are defined as follows.*

$\mathsf{Setup}(1^\kappa)$: *This algorithm takes as input a security parameter* κ *and outputs public parameters* PP*.*

$\mathsf{Gen}(\mathsf{PP})$: *This algorithm takes as input parameters* PP *and outputs a keypair* $(\mathsf{sk}, \mathsf{pk})$*.*

$\mathsf{Sign}(\mathsf{sk}_i, m, \mathcal{R})$: *This algorithm takes as input a secret key* sk_i*, a message* $m \in \mathcal{M}$ *and a ring* $\mathcal{R} = (\mathsf{pk}_j)_{j\in[n]}$ *of* n *public keys such that* $\mathsf{pk}_i \in \mathcal{R}$*. It outputs a signature* σ*.*

$\mathsf{Verify}(m, \sigma, \mathcal{R})$: *This algorithm takes as input a message* $m \in \mathcal{M}$*, a signature* σ *and a ring* $\mathcal{R}$*. It outputs a bit* $b \in \{0, 1\}$*.*

A secure ring signature scheme needs to be correct, unforgeable, and anonymous. While we omit the obvious correctness definition, we provide formal definitions for the remaining properties, following [BKM09], in Definition Collection 1. We note that Bender et al. in [BKM09] have formalized multiple variants of these properties, where we always use the *strongest* one.

4.2 Generic Approaches to Design Ring Signatures

A folklore approach to design ring signatures in the random oracle model is to use the **NP** relation R_{RS} together with a one-way function μ, which defines the relation between secret and public keys:

$$(\mathcal{R}, \mathsf{sk}) \in R_{\mathsf{RS}} \iff \exists\ \mathsf{pk}_i \in \mathcal{R}_{\mathsf{RS}}\ :\ \mathsf{pk}_i = \mu(\mathsf{sk}),$$

and allows to demonstrate knowledge of a witness (a secret key) of one of the public keys in the ring $\mathcal{R}$. Usually, one then designs a Σ-protocol for relation R_{RS} and converts it into a signature scheme using the Fiat-Shamir heuristic.

Linear-Size Signatures. A frequently used instantiation of the above approach is instantiating the relation above by means of a disjunctive proof of knowledge [CDS94]. Using this approach, one obtains ring signatures of linear size. It might be tempting to think that there is a lot of optimization potential for signature sizes in ring signatures. However, without additional assumptions about how the keys are provided to the verifier, signatures of linear size are already the best one can hope for: the verifier needs to get every public key in the ring to verify the signature.

Unforgeability requires that without any secret key sk_i that corresponds to a public key $\mathsf{pk}_i \in \mathcal{R}$, it is infeasible to produce valid signatures with respect to arbitrary such rings $\mathcal{R}$. Our unforgeability notion is the strongest notion defined in [BKM09] and is there called *unforgeability w.r.t. insider corruption.*

Definition 10 (Unforgeability). *A ring signature scheme provides unforgeability, if for all PPT adversaries $\mathcal{A}$, there exists a negligible function $\varepsilon(\cdot)$ such that it holds that*

$$\Pr\left[\begin{array}{l} \mathsf{PP} \leftarrow \mathsf{Setup}(1^\kappa), \\ \{(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\mathsf{PP})\}_{i \in [\mathsf{poly}(\kappa)]}, \\ \mathcal{O} \leftarrow \{\mathsf{Sig}(\cdot,\cdot,\cdot), \mathsf{Key}(\cdot)\}, \\ (m^\star, \sigma^\star, \mathcal{R}^\star) \leftarrow \mathcal{A}^{\mathcal{O}}(\{\mathsf{pk}_i\}_{i \in [\mathsf{poly}(\kappa)]}) \end{array} : \begin{array}{r} \mathsf{Verify}(m^\star, \sigma^\star, \mathcal{R}^\star) = 1 \ \wedge \\ (\cdot, m^\star, \mathcal{R}^\star) \notin \mathcal{Q}^{\mathsf{Sign}} \ \wedge \\ \mathcal{R}^\star \subseteq \{\mathsf{pk}_i\}_{i \in [\mathsf{poly}(\kappa)] \setminus \mathcal{Q}^{\mathsf{Key}}} \end{array}\right] \leq \varepsilon(\kappa),$$

where $\mathsf{Sig}(i, m, \mathcal{R}) := \mathsf{Sign}(\mathsf{sk}_i, m, \mathcal{R})$*,* Sig *returns* $\perp$ *if* $\mathsf{pk}_i \notin \mathcal{R} \ \vee \ i \notin [\mathsf{poly}(\kappa)]$*, and* $\mathcal{Q}^{\mathsf{Sig}}$ *records the queries to* Sig*. Furthermore,* $\mathsf{Key}(i)$ *returns* sk_i *and* $\mathcal{Q}^{\mathsf{Key}}$ *records the queries to* Key*.*

Anonymity requires that it is infeasible to tell which ring member produced a certain signature as long as there are at least two honest members in the ring. Our anonymity notion is the strongest notion defined in [BKM09] and is there called *anonymity against full key exposure.*

Definition 11 (Anonymity). *A ring signature scheme provides anonymity, if for all PPT adversaries $\mathcal{A}$ and for all polynomials* $\mathsf{poly}(\cdot)$*, there exists a negligible function $\varepsilon(\cdot)$ such that it holds that*

$$\Pr\left[\begin{array}{l} \mathsf{PP} \leftarrow \mathsf{Setup}(1^\kappa), \\ \{(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{Gen}(\mathsf{PP})\}_{i \in [\mathsf{poly}(\kappa)]}, \\ b \xleftarrow{R} \{0,1\}, \ \mathcal{O} \leftarrow \{\mathsf{Sig}(\cdot,\cdot,\cdot)\}, \\ (m, j_0, j_1, \mathcal{R}, \mathsf{st}) \leftarrow \mathcal{A}^{\mathcal{O}}(\{\mathsf{pk}_i\}_{i \in [\mathsf{poly}(\kappa)]}), \\ \sigma \leftarrow \mathsf{Sign}(\mathsf{sk}_{j_b}, m, \mathcal{R}), \\ b^\star \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{st}, \sigma, \{\mathsf{sk}_i\}_{i \in [\mathsf{poly}(\kappa)]}) \end{array} : \begin{array}{r} b = b^\star \ \wedge \\ \{\mathsf{pk}_{j_i}\}_{i \in \{0,1\}} \subseteq \mathcal{R} \end{array}\right] \leq 1/2 + \varepsilon(\kappa),$$

where $\mathsf{Sig}(i, m, \mathcal{R}) := \mathsf{Sign}(\mathsf{sk}_i, m, \mathcal{R})$*.*

Definition Collection 1. Security Definitions for Ring Signature Schemes

Reducing Signature Size. However, to further reduce the signature size there is a nice trick which is based on the observation that in many practical scenarios the prospective ring members are already clear prior to the signature generation. Consequently, one can compactly encode all public keys in this ring within some suitable structure and compute the signatures with respect to this compact structure. This trick was first used by Dodis et al. [DKNS04]. Loosely their approach can be described as follows. They use a cryptographic accumulator with a one-way domain to accumulate the ring $\mathcal{R}$, a set of public keys being the output of applying the one-way function μ to the respective secret key. This way they obtain a succinct representation of $\mathcal{R}$. Then, they use a proof system that allows to prove knowledge of a witness of one accumulated value (i.e., the public key) and knowledge of the pre-image thereof (i.e., the corresponding secret key). This proof can be turned into a signature using the Fiat-Shamir heuristic.

Depending on the size of the zero-knowledge membership proof this can yield sublinear (logarithmic or even constant size) signatures. Dodis et al. presented an instantiation of an accumulator together with the respective zero-knowledge proofs that yield constant size ring signatures based on the strong RSA assumption. Logarithmic size ring signatures under lattice assumptions are presented in [LLNW16].

4.3 Our Construction of Logarithmic Size Ring Signatures

Our construction basically follows the approach discussed above to reduce signature size. However, in contrast to Dodis et al., besides targeting the post-quantum setting, we (1) *do not* require a trusted setup[7], and (2) cannot rely on accumulators with one-way domain which provide *quasi-commutativity*. Latter is too restricting and not compatible with the setting in which we work. In particular, it excludes Merkle tree accumulators, which is why we chose to rely on a more generic formalization of accumulators (cf. Section 3). Like Dodis et al., we assume that in practical situations rings often stay the same for a long period of time (e.g., some popular rings are used very often by various members of the ring), or have an implicit short description. Consequently, we measure the signature size as that of the actual signature, i.e., the information one requires *in addition* to the group description. We want to stress once again that when counting the description of the ring as part of the signature, every secure ring signature schemes needs to have signature sizes which are at least linear in the size of the ring.

For the ease of presentation let us fix one such popular ring $\mathcal{R}$ identified by the corresponding accumulator $\Lambda_{\mathcal{R}}$ and we assume that $|\mathcal{R}| = 2^t$ for some $t \in \mathbb{N}$.[8] We present our construction as Scheme 2.

Remark 2. Note that in Scheme 2 crs is not a common reference string (CRS) that needs to be honestly computed by a trusted third party. We simply stick with the notion including a CRS for formal reasons, i.e,. to allow the abstract notion of NIZKs, but as we exclusively use NIZK from Σ-protocols, we do not require a trusted setup and crs is just a description of the hash function which can be globally fixed, e.g., to SHA-256 or SHA-3. Recall, within Fiat-Shamir $\Pi.\mathsf{Setup}(1^\kappa)$ fixes a RO $H : \mathsf{A} \times \mathsf{X} \to \mathsf{E}$, sets $\mathsf{crs} \leftarrow (1^\kappa, H)$ and returns crs.

Remark 3. A trusted setup in context of ring signatures is actually problematic, as it assumes that some mutually trusted party honestly executes the setup. For instance, in case of the strong RSA accumulator [BP97, CL02] as used within [DKNS04], the party running the Gen algorithm of the accumulator can arbitrarily cheat. This can easily be done by keeping the accumulator secret (a trapdoor) instead of discarding it. Using this information, a dishonest setup allows to insert and delete arbitrary elements into and from the accumulator

[7] A trusted setup somehow undermines the idea behind ring signatures.

[8] If this is not the case, one can always add dummy keys to the ring to satisfy this condition.

$\underline{\mathsf{Setup}(1^\kappa)}$: Let Λ be the accumulator with one-way domain based on Scheme 1, run $(\mathsf{sk}_\Lambda, \mathsf{pk}_\Lambda) \leftarrow \Lambda.\mathsf{Gen}(1^\kappa, t)$ (note that $\mathsf{sk}_\Lambda = \emptyset$). Run $\mathsf{crs} \leftarrow \Pi.\mathsf{Setup}(1^\kappa)$ and return $\textsc{pp} \leftarrow (\mathsf{pk}_\Lambda, \mathsf{crs}) = ((H_k, f_\Lambda), (1^\kappa, H))$.

$\underline{\mathsf{KeyGen}(\textsc{pp})}$: Parse $\textsc{pp}$ as $((H_k, f_\Lambda), \mathsf{crs})$, run $(x, z) \leftarrow f_\Lambda.W(1^\kappa)$, and set $\mathsf{pk} \leftarrow (\textsc{pp}, x)$, $\mathsf{sk} \leftarrow (\mathsf{pk}, z)$. Return $(\mathsf{sk}, \mathsf{pk})$.

$\underline{\mathsf{Sign}(\mathsf{sk}_i, m, \mathcal{R})}$: Parse sk_i as $((((H_k, f_\Lambda), \mathsf{crs}), x_i), z_i)$ and $\mathcal{R}$ as $(\mathsf{pk}_1, \ldots, \mathsf{pk}_t) = ((\cdot, x_1), \ldots, (\cdot, x_t))$. Let $\mathcal{X} = (x_1, \ldots, x_t)$, run $(\Lambda_\mathcal{X}, \mathsf{aux}) \leftarrow \Lambda.\mathsf{Eval}((\cdot, \mathsf{pk}_\Lambda), \mathcal{X})$ and $\mathsf{wit}_{f_\Lambda(z_i)} \leftarrow \Lambda.\mathsf{WitCreate}((\cdot, \mathsf{pk}_\Lambda), \Lambda_\mathcal{X}, \mathsf{aux}, f_\Lambda(z_i))$. Obtain $(z_i, (a_j)_{j\in[t]}, (b_j)_{j\in[t]}) \leftarrow \mathsf{Trans}(z_i, \mathsf{wit}_{f_\Lambda(z_i)})$, and return the signature $\sigma \leftarrow (\pi, \Lambda_\mathcal{X})$, where

$$\pi \leftarrow \Pi.\mathsf{Proof}(\mathsf{crs}, (\Lambda_\mathcal{X}, f_\Lambda, H_k), (z_i, (a_j)_{j\in[t]}, (b_j)_{j\in[t]})).$$

$\underline{\mathsf{Verify}(m, \sigma, \mathcal{R})}$: Parse σ as $(\pi, \Lambda_\mathcal{X})$ and $\mathcal{R}$ as $(\mathsf{pk}_1, \ldots, \mathsf{pk}_t) = ((((H_k, f_\Lambda), \mathsf{crs}), x_1), \ldots, (\cdot, x_t))$. Let $\mathcal{X} = (x_1, \ldots, x_t)$, and compute

$$(\Lambda'_\mathcal{X}, \mathsf{aux}') \leftarrow \Lambda.\mathsf{Eval}((\cdot, \mathsf{pk}_\Lambda), \mathcal{X}).$$

If $\Lambda'_X \neq \Lambda_X$ return 0. Otherwise return $\Pi.\mathsf{Verify}(\mathsf{crs}, (\Lambda_\mathcal{X}, f_\Lambda, H_k), \pi)$.

Scheme 2. Construction of logarithmic size RS.

without changing the accumulator value. In context of ring signatures one thus can arbitrarily modify existing rings used within signatures, which could lead to modification of rings to just include public keys into the ring so that for every member of the ring the sole fact to know that one of these persons produced a signature already leads to severe consequences. *We stress that in our case there is no trusted setup. In particular, there is no accumulator secret and the public parameters are just descriptions of hash functions and a OWF.*

Now, we argue that our ring signature presented in Scheme 2 represents a secure ring signature scheme, where we omit correctness which is straightforward to verify.

Theorem 1. *If Λ is a collision free accumulator with one-way domain with respect to f_Λ and Π is a simulation-sound extractable non-interactive proof system, then the ring signature scheme in Scheme 2 is unforgeable.*

Proof. We prove unforgeability using a sequence of games.

Game 0: The original unforgeability game.

Game 1: As Game 0, but we modify Gen to setup (crs, τ) using $\mathcal{S}_1$ and henceforth simulate all proofs in Sign without a witness using τ.

Transition - Game 0 → Game 1: A distinguisher between Game 0 and Game 1 is a zero-knowledge distinguisher for Π, i.e., $|\Pr[S_0] - \Pr[S_1]| \leq \varepsilon_{\mathsf{zk}}(\kappa)$.

Game 2: As Game 1, but we further modify Gen to setup $(\mathsf{crs}, \tau, \xi)$ using $\mathcal{E}_1$ and store ξ.

Transition - Game 1 → Game 2: By simulation-sound extractability, this change is only conceptual, i.e., $\Pr[S_1] = \Pr[S_2]$.

Game 3: As Game 2, but for the forgery $(m^\star, \sigma^\star, \mathcal{R}^\star)$ output by the adversary we parse $\sigma^\star$ as $(\pi, \Lambda_{\mathcal{X}})$ and obtain $(z_i, (a_i)_{i\in[k]}, (b_i)_{i\in[k]}) \leftarrow \mathcal{E}_2(\mathsf{crs}, \xi, (\Lambda_{\mathcal{X}}, f_\Lambda, H_k), \pi)$. If the extractor fails, we abort.

Transition - Game 2 → Game 3: Game 2 and Game 3 proceed identically, unless we abort. The probability for the abort event to happen is upper bounded by $\varepsilon_{\mathsf{ext}}(\kappa)$ which is why we can conclude that $|\Pr[S_3] - \Pr[S_2]| \leq \varepsilon_{\mathsf{ext}}(\kappa)$.

Game 4: As Game 3, but we abort if we have extracted $(z_i, (a_i)_{i\in[n]}, (b_i)_{i\in[n]})$ so that we have that $(\cdot, \mathsf{wit}_{f_\Lambda(z_i)}) \leftarrow \mathsf{Trans}^{-1}(z_i, (a_i)_{i\in[n]}, (b_i)_{i\in[n]})$ is a valid witness for some $f_\Lambda(z_i)$ which was never accumulated.

Transition - Game 3 → Game 4: If we abort in Game 4, we have a collision for the accumulator. That is $|\Pr[S_3] - \Pr[S_4]| \leq \varepsilon_{\mathsf{cf}}(\kappa)$.

Game 5: As Game 4, but we guess the index $i^\star$ the adversary will attack beforehand, and abort if our guess is wrong.

Transition - Game 4 → Game 5: The success probability in Game 4 is the same as in Game 5, unless our guess is wrong, i.e., $\Pr[S_5] = 1/\mathsf{poly}(\kappa) \cdot \Pr[S_4]$.

Game 6: As Game 5, but instead of honestly generating the keypair for user $i^\star$, we engage with a challenger of a OWF to obtain $x_{i^\star}$ and include it in $\mathsf{pk}_{i^\star}$ accordingly. We set $\mathsf{sk}_{i^\star} \leftarrow \emptyset$.

Transition - Game 5 → Game 6: This change is conceptual, i.e., $\Pr[S_5] = \Pr[S_6]$.

In the last game, we have an adversary against the OWF, i.e., $\Pr[S_6] \leq \varepsilon_{\mathsf{owf}}(\kappa)$. All in all, we have that $\Pr[S_0] \leq \mathsf{poly}(\kappa) \cdot \varepsilon_{\mathsf{owf}}(\kappa) + \varepsilon_{\mathsf{zk}}(\kappa) + \varepsilon_{\mathsf{ext}}(\kappa) + \varepsilon_{\mathsf{cf}}(\kappa)$

Theorem 2. *If Π is a zero-knowledge non-interactive proof system, then the ring signature scheme in Scheme 2 is anonymous.*

Proof. We prove anonymity using a sequence of games.

Game 0: The original anonymity game.

Game 1: As Game 0, but we modify Gen to setup (crs, τ) using $\mathcal{S}_1$ and henceforth simulate all proofs in Sign without a witness using τ.

Transition - Game 0 → Game 1: A distinguisher between Game 0 and Game 1 is a zero-knowledge distinguisher for Π, i.e., $|\Pr[S_0] - \Pr[S_1]| \leq \varepsilon_{\mathsf{zk}}(\kappa)$.

In Game 1 the simulation is independent of b, meaning that $\Pr[S_1] = 1/2$. Thus, we have $\Pr[S_0] \leq 1/2 + \varepsilon_{\mathsf{zk}}(\kappa)$, which concludes the proof. □

5 Implementation Aspects and Evaluation

In this section we discuss some implementation aspects regarding instantiating our ring signature scheme. Moreover, we evaluate the efficiency of a concrete instantiation. Since we require simulation-sound extractable NIZK proof systems, we confirm that the Fiat-Shamir (resp. Unruh) transformed version of ZKB++ represents a suitable proof system in the ROM (resp. QROM). We again want to note that we were not able to include the ZKB++ construction due to space limitations, but refer the reader to [CDG+17] for the details.

5.1 Simulation-Sound Extractability of ZKB++

To instantiate our ring signature scheme using ZKB++, we first need to confirm that the NIZK proof system obtained by applying the Fiat-Shamir/Unruh transform to ZKB++ is in fact simulation-sound extractable. For the Unruh-transformed proof system, this was already shown in [CDG+17, Theorem 2] in the QROM, which is why we only focus on the Fiat-Shamir version. We base our argumentation upon the argumentation in [FKMV12]. What we have to do is to show that the FS transformed ZKB++ is zero-knowledge and provides quasi-unique responses in the ROM. We do so by proving two lemmas. Combining those lemmas with [FKMV12, Theorem 2 and Theorem 3] then yields simulation-sound extractability as a corollary.

Lemma 3. *Let Q_H be the number of queries to the random oracle H, Q_S be the overall queries to the simulator, and let the commitments be instantiated via a RO H' with output space $\{0,1\}^\rho$ and the committed values having min entropy ν. Then the probability $\epsilon(\kappa)$ for all PPT adversaries $\mathcal{A}$ to break zero-knowledge of κ parallel executions of the FS transformed* ZKB++ *is bounded by $\epsilon(\kappa) \leq {}^{s}/{}_{2^\nu} + {}^{(Q_S \cdot Q_H)}/{}_{2^{3 \cdot \rho}}$.*

The lemma above was already proven for ZKBoo in [DOR+16]. For ZKB++ the argumentation is the same. We restate the proof below for completeness.

Proof. We bound the probability of any PPT adversary $\mathcal{A}$ to win the zero-knowledge game by showing that the simulation of the proof oracle is statistically close to the real proof oracle. For our proof let the environment maintain a list $\mathtt{H}$ where all entries are initially set to $\perp$.

Game 0: The zero-knowledge game where the proofs are honestly computed, and the ROs are simulated honestly.

Game 1: As Game 0, but whenever the adversary requests a proof for some tuple (x, w) we choose $\mathsf{e} \xleftarrow{R} \{0,1,2\}^\kappa$ before computing a and z. If $\mathtt{H}[(\mathsf{a}, x)] \neq \perp$ we abort and call that event E. Otherwise, we set $\mathtt{H}[(\mathsf{a}, x)] \leftarrow \mathsf{e}$.

Transition - Game 0 → Game 1: Both games proceed identically unless E happens. The message a includes 3 RO commitments with respect to H', i.e., the min-entropy is lower bounded by $3 \cdot \rho$. We have $|\Pr[S_0] - \Pr[S_1]| \leq {}^{(Q_S \cdot Q_H)}/{}_{2^{3 \cdot \rho}}$.

Game 2: As Game 1, but we compute the commitments in a so that the ones which will never be opened according to e contain random values.

Transition - Game 1 → Game 2: The statistical difference between Game 1 and Game 2 can be upper bounded by $|\Pr[S_1] - \Pr[S_2]| \leq \kappa \cdot {}^{1}/{}_{2^\nu}$ (for compactness we collapsed the s game changes into a single game).

Game 3: As Game 2, but we use the HVZK simulator to obtain $(\mathsf{a}, \mathsf{e}, \mathsf{z})$.

Transition - Game 2 → Game 3: This change is conceptual, i.e., $\Pr[S_2] = \Pr[S_3]$.

In Game 0, we sample from the first distribution of the zero-knowledge game, whereas we sample from the second one in Game 3; the distinguishing bounds shown above conclude the proof. □

Lemma 4. *Let the commitments be instantiated via a RO H' with output space $\{0,1\}^\rho$ and let $Q_{H'}$ be the number of queries to H', then the probability to break quasi-unique responses is bounded by $^{Q_{H'}^2}/_{2^\rho}$.*

Proof. To break quasi-unique responses, the adversary would need to come up with two valid proofs $(\mathsf{a}, \mathsf{e}, \mathsf{z})$ and $(\mathsf{a}, \mathsf{e}, \mathsf{z}')$. The last message z (resp z') only contains openings to commitments, meaning that breaking quasi unique responses implies finding a collision for at least one of the commitments. The probability for this to happen is upper bounded by $^{Q_{H'}^2}/_{2^\rho}$ which concludes the proof. □

Combining Lemma 3 and Lemma 4 with [FKMV12, Theorem 2 and Theorem 3] yields the following corollary.

Corollary 1. *The FS transformed* ZKB++ *is simulation-sound extractable.*

5.2 Implementation of the Circuit

One very important factor, when it comes to the actual size of the signatures, is the concrete implementation of the circuit. To this end, we explicitly describe our design strategy, which uses 2-to-1 multiplexers as central components. A 2-to-1 multiplexer selects between two input wires based on a selection bit. In particular, given a selection bit s and two input wires i_0 and i_1, the multiplexer outputs i_s. More formally, we can describe the multiplexer as function ν defined as

$$\nu(s, i_0, i_1) = (\neg s \ \wedge \ i_0) \ \vee \ (s \ \wedge \ i_1).$$

This function can be expressed using only 1 AND and 2 XOR gates for 1-bit input wires as

$$\nu(s, i_0, i_1) = ((i_0 \oplus i_1) \wedge s) \oplus i_0.$$

For values of multiple bits, ν is applied bit-by-bit. Now, to instantiate $a_i = H_k(a_{i+1} \| b_{i+1}) \ \vee \ a_i = H_k(b_{i+1} \| a_{i+1})$ in a straight-forward way, it would be possible to write the expression as $a_i = \nu(s_{i+1}, H_k(a_{i+1} \| b_{i+1}), H_k(b_{i+1} \| a_{i+1})$ where the selection bits s_{i+1} encode the path and are additionally part of the witness. Besides the cost for the two hash function evaluations, this would require us to account for 1 AND gate for each output bit of the hash function H_k and each level of the Merkle tree accumulator.

However, with a little more care, we can even obtain a more efficient solution. Namely, when shifting the multiplexer from the output of the hash function to its inputs and when additionally rotating a_{i+1} and b_{i+1} based on the selection bit we obtain an equivalent representation of the equation given above:

$$a_i = H_k(\nu(s_{i+1}, a_{i+1}, b_{i+1}) \| \nu(s_{i+1}, b_{i+1}, a_{i+1})).$$

When we let $\nu'(s, i_0, i_1) = \nu(s, i_0, i_1) \| \nu(s, i_1, i_0)$, we can write this as

$$a_i = H_k(\nu'(s_{i+1}, a_{i+1}, b_{i+1})).$$

The important point to observe here is that we can trade one (cheap) additional multiplexer for one (expensive) full evaluation of the hash function. Note that ν' computes $(i_0 \oplus i_1) \wedge s$ twice, and thus we can further simplify it and explicitly instantiate ν' as conditional swap gate:

$$\nu'(s, i_0, i_1) = i_0 \oplus s' \| i_1 \oplus s' \text{ where } s' = (i_0 \oplus i_1) \wedge s.$$

Thus ν' only requires 1 AND gate. Additionally, the possibility to swap the inputs allows us to drop all a_i from the witness and to re-write the statement as

$$\Lambda_{\mathcal{X}} = H_k(\nu'(s_1, H_k(\ldots H_k(\nu'(s_k, f_\Lambda(z), b_k), \ldots), b_1))).$$

We remark that this statement bears similarities with the recent statement used by Boneh et al. [BEF18]. However, in contrast to our work, they introduce a novel property for the used hash function which they term third preimage resistance, and construct a third preimage resistant hash function H as $H(x, y) = H(x, y) \oplus H(y, x)$. The important difference to our approach is that they need two evaluations of the hash function, whereas we only need one.

5.3 Selection of Symmetric-Key Primitives

When instantiating our ring signature scheme using ZKB++, the selection of the underlying primitives is of importance for the actual signature sizes as well as the overall performance. As ZKB++'s proof size depends on the number of multiplication gates and the size of the operands, we require a OWF and a collision-resistant hash function with a representation as circuit, where the product of the multiplicative complexity and the number of bits required to store field elements is minimal. Note that for the OWF we can observe that, when instantiating it with a block cipher, only one plaintext-ciphertext pair per key is visible to an adversary. Hence, we have the same requirements as in [CDG+17], which is why we also choose LowMC [ARS+15, ARS+16] with a reduced data complexity to build the OWF. For the selection of the collision-resistant hash function we are presented with different options:

Standardized Hash Functions. SHA-256 or SHA-3 are the obvious choices for collision resistant hash functions. SHA-256's compression function requires around 25000 multiplication gates [BCG+14] and SHA-3's permutation even more with around 38400 gates [NIS15].

Sponge Construction with Low Multiplicative Complexity Ciphers. Using a block cipher with small multiplicative complexity as permutation in a sponge construction, e.g., using LowMC or MiMC [AGR+16], enables the construction of hash functions with similar security guarantees as SHA-256 and SHA-3, but with a significantly reduced multiplicative complexity. Using the numbers from [AGR+16], MiMCHash-256 requires 1293 multiplications with a field size of 1025 bits. LowMCHash-256 only requires a 1 bit binary field and 3540 AND gates[9]. Thus, a hash based on LowMC is a better candidate for our use case.

[9] Numbers updated according to a personal discussion with Christian Rechberger.

Davies-Mayer Transformation with Low Multiplicative Complexity Ciphers. As a lower-cost alternative to sponge based constructions, Boneh et al. [BEF18] suggest to use a collision resistant hash function obtained by applying the Davis-Meyer transformation to a block cipher with low multiplicative complexity. This is reasonable because collision resistance is only required for a fixed message length, being equivalent to the sum of block size and key size of the underlying blockcipher (i.e., LowMC).

5.4 Signature Size Estimations

Finally we present signature sizes when instantiating our ring signature scheme with LowMC for both, the OWF and the hash function. For the instantiation of the hash function we present estimations based on sponge constructions as well as Davies-Mayer constructions. Table 1 presents the signature size estimations for the sponge-based instantiation for different choices of ring sizes and aiming at a 128 bit post-quantum security level. We compute them using the formulas from [CDG+17]. The proofs are of size $t \cdot (c + 2s + \log_2(3) + \ell \cdot m + i))$ bits when using the Fiat-Shamir transform, and of $t \cdot (c + 3s + \log_2(3) + 2\ell \cdot m + i)$ bits when using the Unruh-transform, respectively, where t is the number of repetitions, c the size of the commitments, i the size of the input to the circuit, ℓ the size of the underlying field, m the number of AND gates, and s the size of the seeds used to generate the random tapes. We use ZKB++ as instantiated in [CDG+17] and give the numbers for both the Fiat-Shamir and Unruh transformed proof system.

Ring size	$\lvert\sigma\rvert$ (FS/ROM)	$\lvert\sigma\rvert$ (Unruh/QROM)
2^k	$948708 + 1775214 \cdot k$ bits	$1560156 + 3437862 \cdot k$ bits
2^5	1200 KB	2289 KB
2^{10}	2283 KB	4388 KB
2^{20}	4450 KB	8584 KB

Table 1. Signature sizes at the 128 bit post-quantum security level using LowMC with 1024 bit block size, 10 S-boxes and 118 rounds in the sponge framework.

Following [BEF18] we also present numbers using a hash function obtained using the Davies-Meyer transform in Table 2.

Ring size	$\lvert\sigma\rvert$ (FS/ROM)	$\lvert\sigma\rvert$ (Unruh/QROM)
2^k	$948708 + 986814 \cdot k$ bits	$1560156 + 1861062 \cdot k$ bits
2^5	719 KB	1327 KB
2^{10}	1321 KB	2463 KB
2^{20}	2526 KB	4735 KB

Table 2. Signature sizes at the 128 bit post-quantum security level using Davies-Meyer with LowMC with 256 bit block size, 10 S-boxes and 58 rounds.

Improvements compared to the Work of Boneh et al. [BEF18]. Using the same LowMC instance with 1374 AND gates as Boneh et al. [BEF18], we obtain significantly shorter sizes for the membership proof in the accumulator (and therefore also significantly shorter ring signature sizes) compared to what we would obtain when using their techniques to prove membership in the accumulator in zero knowledge. For ring size of 2^{20} we obtain 2134 KB when using Fiat-Shamir and 3952 KB when using Unruh, respectively. This reduction in proof (signature) size is not surprising: observe that the statement they have to prove requires the evaluation of two hash functions per level in the tree. We only require a conditional swap gate and a single hash function evaluation yielding a much lower number of required AND gates (roughly $1/2$). We want to stress that our conditional swap gate-based approach is also useful to reduce the group signature sizes of [BEF18, Construction II], which internally uses zero-knowledge membership proofs with respect to a Merkle tree accumulator.

Finally, we also note that Ligero [AHIV17], a recent NIZK proof system for general circuits, offers proofs of logarithmic size in the number of multiplication gates in the prime field case respectively in the number of AND and XOR gates in the case of binary fields, which would allow us to reduce the signature size significantly. However, to the best of our knowledge, it is unclear whether Ligero provides simulation-sound extractability.

6 Conclusions

In this this work we made some important steps towards establishing privacy-enhancing primitives which are solely built from symmetric-key primitives and therefore do not require any structured hardness assumptions. In our work, we followed a modular concept and first introduced a post-quantum accumulator with efficient zero-knowledge membership proofs of sublinear size. Besides the applications to logarithmic size ring signatures as we presented in this paper, we believe that our post-quantum accumulator construction with zero-knowledge proofs may well have broader impact in the construction of other (privacy-enhancing) protocols in the post-quantum setting.

Open Questions. In addition, we believe that our work also opens up quite some possibilities for further research.

First, in the context of privacy-enhancing protocols, it would be interesting to investigate how to extend our methods to obtain group signatures [CvH91], i.e., anonymous signatures that provide the possibility to re-identify anonymous signers by a dedicated party. We note that Dodis et al. [DKNS04] informally discuss that when adding ID escrow functionality to their ring signature scheme yields group signatures. Basically, the lattice-based construction of Libert et al. [LLNW16] can be considered as an instantiation of the former paradigm. The problem is that this paradigm requires IND-CCA2 secure public-key encryption, which does not exist given our constraints. In addition, it is well known [AW04, CG04] that group signatures in the static model by Bellare et al. in [BMW03] imply public-key encryption. This means that the best one could hope for would be a construction being secure in a weakened version of the Bellare et al. model. Work in this direction was earlier pursued by Camenisch and Groth [CG04], who showed how to construct group signature schemes in a weaker model from one-way functions and non-interactive zero-knowledge arguments. The question which remains open in our context is whether one can find instantiations without the requirement for structured hardness assumptions and providing the practical efficiency one would hope for, i.e., ideally instantiations which just require to prove statements with respect to a few evaluations of a block cipher. A similar question was recently investigated by Boneh et al. in [BEF18], where they constructed practical group signature schemes from symmetric-key primitives. They use a security model without opening mechanism but with revocation feature for keys and signatures, respectively. Since this is a different model, our question still remains open.

Second, in the context of symmetric-key primitives, one may observe that—despite the recent trend to construct symmetric-key primitives with particularly low AND count—there is no practical application so far which would require collision resistant hash functions with particularly low AND count. Since our accumulator construction relies on collision resistant hash functions, our work may well also open up new fields of research in the symmetric-key community.

Acknowledgments. The authors have been supported by EU H2020 Project Prismacloud, grant agreement n°644962. We thank Christian Rechberger for discussions on the choice of symmetric-key primitives, especially regarding the instantiation of hash functions using LowMC, as well as for providing us with updated LowMC instances. We also thank Daniel Kales for ideas to reduce the cost for the conditional swap gate.

References

[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 *ASIACRYPT (1)*, volume 10031 of *Lecture Notes in Computer Science*, pages 191–219, 2016.

[AHIV17] S. Ames, C. Hazay, Y. Ishai, and M. Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In *ACM Conference on Computer and Communications Security*, pages 2087–2104. ACM, 2017.

[ARS$^+$15] M. R. Albrecht, C. Rechberger, T. Schneider, T. Tiessen, and M. Zohner. Ciphers for MPC and FHE. In *EUROCRYPT (1)*, volume 9056 of *Lecture Notes in Computer Science*, pages 430–454. Springer, 2015.

[ARS$^+$16] M. R. Albrecht, C. Rechberger, T. Schneider, T. Tiessen, and M. Zohner. Ciphers for MPC and FHE. *IACR Cryptology ePrint Archive*, 2016:687, 2016.

[AW04] M. Abdalla and B. Warinschi. On the minimal assumptions of group signature schemes. In *ICICS*, volume 3269 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2004.

[BCC04] E. F. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *ACM Conference on Computer and Communications Security*, pages 132–145. ACM, 2004.

[BCC$^+$15] J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, J. Groth, and C. Petit. Short accountable ring signatures based on DDH. In *ESORICS (1)*, volume 9326 of *Lecture Notes in Computer Science*, pages 243–265. Springer, 2015.

[BCD+17] F. Baldimtsi, J. Camenisch, M. Dubovitskaya, A. Lysyanskaya, L. Reyzin, K. Samelin, and S. Yakoubov. Accumulators with applications to anonymity-preserving revocation. In *EuroS&P*, pages 301–315. IEEE, 2017.

[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 *IEEE Symposium on Security and Privacy*, pages 459–474. IEEE Computer Society, 2014.

[BdM93] J. C. Benaloh and M. de Mare. One-way accumulators: A decentralized alternative to digital sinatures (extended abstract). In *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pages 274–285. Springer, 1993.

[BEF18] D. Boneh, S. Eskandarian, and B. Fisch. Post-quantum group signatures from symmetric primitives. *IACR Cryptology ePrint Archive*, 2018:261, 2018.

[BKM09] A. Bender, J. Katz, and R. Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. *J. Cryptology*, 22(1):114–138, 2009.

[BL07] E. Brickell and J. Li. Enhanced privacy id: a direct anonymous attestation scheme with enhanced revocation capabilities. In *WPES*, pages 21–30. ACM, 2007.

[BLL00] A. Buldas, P. Laud, and H. Lipmaa. Accountable certificate management using undeniable attestations. In *ACM Conference on Computer and Communications Security*, pages 9–17. ACM, 2000.

[BMW03] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–629. Springer, 2003.

[BP97] N. Baric and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *EUROCRYPT*, volume 1233 of *Lecture Notes in Computer Science*, pages 480–494. Springer, 1997.

[BPW12] D. Bernhard, O. Pereira, and B. Warinschi. How not to prove yourself: Pitfalls of the fiat-shamir heuristic and applications to helios. In *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 626–643. Springer, 2012.

[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 *ACM Conference on Computer and Communications Security*, pages 1825–1842. ACM, 2017.

[CDS94] R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 1994.

[CG04] J. Camenisch and J. Groth. Group signatures: Better efficiency and new theoretical aspects. In *SCN*, volume 3352 of *Lecture Notes in Computer Science*, pages 120–133. Springer, 2004.

[CGS07] N. Chandran, J. Groth, and A. Sahai. Ring signatures of sub-linear size without random oracles. In *ICALP*, volume 4596 of *Lecture Notes in Computer Science*, pages 423–434. Springer, 2007.

[CL02] J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76. Springer, 2002.

[CvH91] D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer, 1991.

[Dam10] I. Damgård. On Σ-protocols. 2010. http://www.cs.au.dk/~ivan/Sigma.pdf.

[DHS15] D. Derler, C. Hanser, and D. Slamanig. Revisiting cryptographic accumulators, additional properties and relations to other primitives. In *CT-RSA*, volume 9048 of *Lecture Notes in Computer Science*, pages 127–144. Springer, 2015.

[DKNS04] Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. Anonymous identification in ad hoc groups. In *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 609–626. Springer, 2004.

[DOR$^+$16] D. Derler, C. Orlandi, S. Ramacher, C. Rechberger, and D. Slamanig. Digital signatures from symmetric-key primitives. *IACR Cryptology ePrint Archive*, 2016:1085, 2016.

[DS16] D. Derler and D. Slamanig. Key-homomorphic signatures and applications to multiparty signatures. *IACR Cryptology ePrint Archive*, 2016:792, 2016.

[FKMV12] S. Faust, M. Kohlweiss, G. A. Marson, and D. Venturi. On the non-malleability of the fiat-shamir transform. In *INDOCRYPT*, volume 7668 of *Lecture Notes in Computer Science*, pages 60–79. Springer, 2012.

[FS86] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[GK15] J. Groth and M. Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In *EUROCRYPT (2)*, volume 9057 of *Lecture Notes in Computer Science*, pages 253–280. Springer, 2015.

[GMO16] I. Giacomelli, J. Madsen, and C. Orlandi. Zkboo: Faster zero-knowledge for boolean circuits. In *USENIX Security Symposium*, pages 1069–1083. USENIX Association, 2016.

[Gon17] A. González. A ring signature of size Θ(sqrt[3]{n}) without random oracles. *IACR Cryptology ePrint Archive*, 2017:905, 2017.

[IKOS09] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Zero-knowledge proofs from secure multiparty computation. *SIAM J. Comput.*, 39(3):1121–1152, 2009.

[LLNW16] B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In *EUROCRYPT (2)*, volume 9666 of *Lecture Notes in Computer Science*, pages 1–31. Springer, 2016.

[MCG08] C. A. Melchor, P. Cayrel, and P. Gaborit. A new efficient threshold ring signature scheme based on coding theory. In *PQCrypto*, volume 5299 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2008.

[MGGR13] I. Miers, C. Garman, M. Green, and A. D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *IEEE Symposium on Security and Privacy*, pages 397–411. IEEE Computer Society, 2013.

[MP17] M. S. E. Mohamed and A. Petzoldt. Ringrainbow - an efficient multivariate ring signature scheme. In *AFRICACRYPT*, volume 10239 of *Lecture Notes in Computer Science*, pages 3–20, 2017.

[MS17] G. Malavolta and D. Schröder. Efficient ring signatures in the standard model. In *ASIACRYPT (2)*, volume 10625 of *Lecture Notes in Computer Science*, pages 128–157. Springer, 2017.

[NIS15] NIST. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. National Institute of Standards and Technology (NIST), FIPS PUB 202, U.S. Department of Commerce, 2015.

[RST01] R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001.

[Unr12] D. Unruh. Quantum proofs of knowledge. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 135–152. Springer, 2012.

[Unr15] D. Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *EUROCRYPT (2)*, volume 9057 of *Lecture Notes in Computer Science*, pages 755–784. Springer, 2015.

[Unr16] D. Unruh. Computationally binding quantum commitments. In *EUROCRYPT (2)*, volume 9666 of *Lecture Notes in Computer Science*, pages 497–527. Springer, 2016.

7

Generic Double-Authentication Preventing Signatures and a Post-Quantum Instantiation

Publication Data

David Derler, Sebastian Ramacher, and Daniel Slamanig. Generic Double-Authentication Preventing Signatures and a Post-quantum Instantiation. In Joonsang Baek, Willy Susilo, and Jongkil Kim, editors, *Provable Security - 12th International Conference, ProvSec 2018, Jeju, South Korea, October 25-28, 2018, Proceedings*, volume 11192 of *Lecture Notes in Computer Science*, pages 258–276. Springer, 2018. URL: https://doi.org/10.1007/978-3-030-01446-9_15

The appended paper is an author-created full version available at https://eprint.iacr.org/2018/790.

Contributions

- The author is one of the main authors.

Generic Double-Authentication Preventing Signatures and a Post-Quantum Instantiation*

David Derler[1], Sebastian Ramacher[1], and Daniel Slamanig[2]

[1] IAIK, Graz University of Technology, Austria
[2] AIT Austrian Institute of Technology GmbH, Vienna, Austria
cryptsec@derler.info, sebastian.ramacher@tugraz.at,
daniel.slamanig@ait.ac.at

Abstract. Double-authentication preventing signatures (DAPS) are a variant of digital signatures which have received considerable attention recently (Derler et al. EuroS&P 2018, Poettering Africacrypt 2018). They are unforgeable signatures in the usual sense and sign messages that are composed of an address and a payload. Their distinguishing feature is the property that signatures on *two different* payloads with respect to the *same* address allow to publicly extract the secret signing key. Thus, they are a means to disincentivize double-signing and are a useful tool in various applications.

DAPS are known in the factoring, the discrete logarithm and the lattice setting. The majority of the constructions are ad-hoc. Only recently, Derler et al. (EuroS&P 2018) presented the first generic construction that allows to extend *any* discrete logarithm based secure signature scheme to DAPS. However, their scheme has the drawback that the number of potential addresses (the address space) used for signing is polynomially bounded (and in fact small) as the size of secret and public keys of the resulting DAPS are linear in the address space. In this paper we overcome this limitation and present a generic construction of DAPS with constant size keys and signatures. Our techniques are not tailored to a specific algebraic setting and in particular allow us to construct the first DAPS without structured hardness assumptions, i.e., from symmetric key primitives, yielding a candidate for post-quantum secure DAPS.

Keywords: digital signatures, double-authentication prevention, Shamir secret sharing, provable-security, generic construction, exponential size address space

1 Introduction

Digital signatures are an important cryptographic primitive used to provide strong integrity and authenticity guarantees for digital messages. Among many

* All authors have been supported by EU H2020 project Prismacloud, grant agreement n°644962. This is the full version of a paper which appears in Provable Security - 12th International Conference, ProvSec 2018, Jeju, South Korea, October 25-28, 2018, Proceedings. ©Springer 2018. The final publication is available via Springer at https://doi.org/10.1007/978-3-030-01446-9_15.

other applications, they are used to issue digital certificates for public keys within public-key infrastructures, to guarantee the origin of executable code, to sign digital documents such as PDF documents (in a legally binding way), as well as in major cryptographic protocols such as TLS. Recently, signatures also emerged to be a cornerstone of distributed cryptocurrencies such as Bitcoin, i.e., are used to bind coins to users (by means of public keys) and to sign transactions.

Double-authentication preventing signatures (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. They provide unforgeability guarantees in the sense of conventional signatures but have the special property that signing two different payloads $p \neq p'$ using the same address a allows to publicly extract the secret signing key from the respective signatures. In the literature, various compelling applications for DAPS have been proposed. Those applications include penalizing double spending attacks in cryptocurrencies [RKS15] or penalizing certification authorities for issuing two certificates with respect to the same domain name, but for two different public keys [PS14], for example. In this work we purely focus on DAPS constructions and we refer the reader to [PS14,PS17] for a comparison with other types of self-enforcing digital signatures.

Currently, DAPS are known in the factoring [PS14,PS17,BPS17], the discrete logarithm [RKS15,DRS18b,Poe18] and the lattice setting [BKN17]. The majority of the constructions (the only exception being [DRS18b]) are ad-hoc. Unfortunately, such an approach yields very specific constructions, whose security may not be well understood. Having generic DAPS constructions, in contrast, yields much more flexibility, as it allows to plug in building blocks whose security is well understood. In addition, this yields simplicity and modularity in the security analysis. Only recently, Derler et al. (EuroS&P 2018) presented the first generic construction that allows to extend *any* discrete logarithm based EUF-CMA secure signatures scheme to DAPS. However, their scheme has the drawback that the number of potential addresses (the address space) used for signing is polynomially bounded (and in fact small) as the size of secret and the public keys of the resulting DAPS are linear in the address space. We ask whether we can come up with a generic construction without this drawback.

Somewhat orthogonal to the motivational discussion above, our work is also driven by the question whether it is possible to construct DAPS without relying on structured hardness assumptions, i.e., solely from symmetric key primitives (following up on a very recent line of work [CDG$^+$17a,DRS18a,BEF18,KKW18]). This is interesting, because symmetric key primitives are conjectured to remain secure in the advent of sufficiently powerful quantum computers. Such quantum computers would break all discrete log and RSA based public key cryptosystems [Sho97].

1.1 Existing DAPS Constructions

DAPS have been introduced by Poettering and Stebila [PS14,PS17] in a factoring-based setting. Ruffing, Kate and Schröder later introduced the notion of accountable assertions (AS) in [RKS15], being a related but weaker primitive than

Approach	Address space	Extraction	Setting	Generic
[PS14,PS17]	exponential	DSE	factoring	×
[RKS15]	exponential	DSE	DLOG	×
[BPS17]	exponential	DSE	factoring	×
[BKN17]	exponential	DSE	lattices	×
[DRS18b]	small	wDSE*	DLOG	✓
[Poe18]	small	DSE	DLOG	×
Construction 1	exponential	wDSE	symmetric	✓
Construction 2	exponential	DSE	any	✓

Table 1: Overview of DAPS constructions

DAPS. In addition they present one AS that also is a DAPS (RKS henceforth). The RKS construction is based on Merkle tress and chameleon hash functions in the discrete logarithm setting. Very recently, Bellare, Poettering and Stebila [BPS17] proposed new factoring-based DAPS from trapdoor identification-schemes using an adaption and extension of a transform from [BPS16]. Their two transforms applied to the Guillou-Quisquater (GQ) [GQ88] and Micali-Reyzin (MR) [MR02] identification scheme yield signing and verification times as well as signature sizes comparable (or slightly above) standard RSA signatures. Boneh et al. [BKN17] propose constructions of DAPS from lattices. They consider DAPS as a special case of what they call predicate-authentication-preventing signatures (PAPS). In PAPS one considers a k-ary predicate on the message space and given any k valid signatures that satisfy the predicate reveal the signing key. Consequently, DAPS are PAPS for a specific 2-ary predicate. Derler, Ramacher and Slamanig (DRS henceforth) in [DRS18b] recently provided the first black-box construction of DAPS from digital signatures schemes and demonstrate how this approach can be used to construct N-times-authentication-preventing signatures (NAPS) (a notion called k-way DAPS in [BKN17]). In addition, they introduced weaker extraction notions, where the focus of the extraction is on the signing key of the underlying signature scheme only. A drawback of their work is that the constructions have $O(n)$ secret and public key size where n is the size of the address space. So their constructions are only suitable for small message spaces. In a follow up work Poettering [Poe18], also focusing on DAPS for small address spaces, showed how for a certain class of signature schemes (obtained via Fiat-Shamir from certain identification schemes), the DRS approach can be improved by reducing the signature size by a factor of five and the size of the secret key from $O(n)$ to $O(1)$. However, this comes at the cost of no longer being able to do a black-box reduction to the underlying signature scheme. In Table 1 we provide a comparison of existing DAPS approaches with the ones presented in this paper regarding address space, extraction capabilities, algebraic setting as well as their characteristic as either being tailored to a specific setting or generic.

1.2 Contribution

Our contributions can be summarized as follows:

- We propose a generic DAPS, respectively NAPS, construction building upon DRS' secret-sharing approach, which resolves the address-space limitation in the DRS construction, and, in particular, supports an exponentially large address space. This improvement is achieved by deriving the coefficients of the secret sharing polynomial from the address using a carefully chosen pseudorandom function with an output domain being compatible with the secret key space of the underlying signature scheme. Consequently, the overhead in the public-key reduces to a constant factor. Like the DRS approach, our generic approach satisfies a relaxed notion of extractability. Interestingly, we can instantiate this construction solely from symmetric-key primitives, yielding a candidate for post-quantum secure DAPS/NAPS.
- While the aforementioned construction thus closes an important gap in the literature, the signature sizes are somewhat large compared to signatures in the discrete log or RSA setting. To this end, we additionally follow a different direction which basically targets the extension of any digital signature scheme (such as ECDSA or EdDSA, for example) to a DAPS. Essentially, we present a compiler which uses an arbitrary DAPS scheme to extend any given signature scheme to a DAPS. While this might sound somewhat odd at first sight, we want to stress that all existing DAPS which have compact keys and exponentially large address space are ad-hoc constructions, whereas practical applications most likely will use standardized signature schemes. Using our construction it is possible to generically bring extraction to any signature scheme. Hence we obtain more efficient DAPS being compatible with standardized signature schemes such as ECDSA or EdDSA.

2 Preliminaries

In this section we firstly present a formal model for the security of signature and DAPS schemes, recall non-interactive zero-knowledge proof systems and Shamir's secret sharing.

2.1 Digital Signature Schemes

Subsequently we formally recall the notion of digital signature schemes.

Definition 1 (Signature Scheme). *A signature scheme* Σ *is a triple* $(\mathsf{KGen}_\Sigma, \mathsf{Sign}_\Sigma, \mathsf{Verify}_\Sigma)$ *of PPT algorithms, which are defined as follows:*

$\mathsf{KGen}_\Sigma(1^\kappa)$: *This algorithm takes a security parameter* κ *as input and outputs a secret (signing) key* sk_Σ *and a public (verification) key* pk_Σ *with associated message space* $\mathcal{M}$ *(we may omit to make the message space* $\mathcal{M}$ *explicit).*

$\mathsf{Sign}_\Sigma(\mathsf{sk}_\Sigma, m)$: *This algorithm takes a secret key* sk_Σ *and a message* $m \in \mathcal{M}$ *as input and outputs a signature* σ*.*

$\mathsf{Verify}_\Sigma(\mathsf{pk}_\Sigma, m, \sigma)$: *This algorithm takes a public key* pk_Σ*, a message* $m \in \mathcal{M}$ *and a signature* σ *as input and outputs a bit* $b \in \{0, 1\}$.

We require a signature scheme to be correct and to provide existential unforgeability under adaptively chosen message attacks ($\mathsf{EUF\text{-}CMA}$ security). For correctness we require that for all $\kappa \in \mathbb{N}$, for all $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow \mathsf{KGen}_\Sigma(1^\kappa)$ and for all $m \in \mathcal{M}$ it holds that

$$\Pr\left[\mathsf{Verify}_\Sigma(\mathsf{pk}_\Sigma, m, \mathsf{Sign}_\Sigma(\mathsf{sk}_\Sigma, m)) = 1\right] = 1.$$

Definition 2 **($\mathsf{EUF\text{-}CMA}$).** *For a PPT adversary* $\mathcal{A}$*, we define the advantage function in the sense of* $\mathsf{EUF\text{-}CMA}$ *as*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\Sigma}(\kappa) = \Pr\left[\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\Sigma}(\kappa) = 1\right]$$

where the corresponding experiment is depicted in Figure 1. If for all PPT adversaries $\mathcal{A}$ *there is a negligible function* $\varepsilon(\cdot)$ *such that*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\Sigma}(\kappa) \leq \varepsilon(\kappa)$$

we say that Σ *is* $\mathsf{EUF\text{-}CMA}$ *secure.*

```
$\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\Sigma}(\kappa)$:
  $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow \mathsf{KGen}_\Sigma(1^\kappa)$
  $\mathcal{Q} \leftarrow \emptyset$
  $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}'_\Sigma(\mathsf{sk}_\Sigma, \cdot)}(\mathsf{pk})$
    where oracle $\mathsf{Sign}'_\Sigma$ on input $m$:
      $\sigma \leftarrow \mathsf{Sign}_\Sigma(\mathsf{sk}_\Sigma, m)$, $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
      return $\sigma$
  return 1, if $\mathsf{Verify}_\Sigma(\mathsf{pk}_\Sigma, m^*, \sigma^*) = 1 \ \wedge \ m^* \notin \mathcal{Q}$
  return 0
```

Fig. 1: $\mathsf{EUF\text{-}CMA}$ **security.**

2.2 Double-Authentication-Preventing Signatures

Double-authentication-preventing signatures (DAPS) are signature schemes being capable of signing messages from a message space $\mathcal{M}$ of the form $\mathsf{A} \times \mathsf{P}$. Each message $m = (a, p) \in \mathcal{M}$ thereby consists of an address a in address space A and a payload p from payload space P. In addition to the algorithms provided by conventional signature schemes, a DAPS scheme provides a fourth algorithm Ex_D that extracts the secret key from signatures on two colliding messages, i.e., two different messages sharing the same address. Formally, a pair of colliding messages is defined as follows:

Definition 3 (Colliding Messages). *We call two messages* $m_1 = (a_1, p_1)$ *and* $m_2 = (a_2, p_2)$ *colliding if* $a_1 = a_2$*, but* $p_1 \neq p_2$*.*

Below, we now formally define DAPS following [PS14,PS17].

Definition 4 (DAPS). *A double-authentication-preventing signature scheme* DAPS *is a tuple* $(\mathsf{KGen_D}, \mathsf{Sign_D}, \mathsf{Verify_D}, \mathsf{Ex_D})$ *of PPT algorithms, which are defined as follows:*

$\mathsf{KGen_D}(1^\kappa)$: *This algorithm takes a security parameter* κ *as input and outputs a secret (signing) key* $\mathsf{sk_D}$ *and a public (verification) key* $\mathsf{pk_D}$ *with associated message space* $\mathcal{M}$ *(we may omit to make the message space* $\mathcal{M}$ *explicit).*

$\mathsf{Sign_D}(\mathsf{sk_D}, m)$: *This algorithm takes a secret key* $\mathsf{sk_D}$ *and a message* $m \in \mathcal{M}$ *as input and outputs a signature* σ*.*

$\mathsf{Verify_D}(\mathsf{pk_D}, m, \sigma)$: *This algorithm takes a public key* $\mathsf{pk_D}$*, a message* $m \in \mathcal{M}$ *and a signature* σ *as input and outputs a bit* $b \in \{0, 1\}$*.*

$\mathsf{Ex_D}(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2)$: *This algorithm takes a public key* $\mathsf{pk_D}$*, two colliding messages* m_1 *and* m_2 *and signatures* σ_1 *for* m_1 *and* σ_2 *for* m_2 *as inputs and outputs a secret key* $\mathsf{sk_D}$*.*

Note that the algorithms $\mathsf{KGen_D}$, $\mathsf{Sign_D}$, and $\mathsf{Verify_D}$ match the definition of the algorithms of a conventional signature scheme. For DAPS one requires a restricted but otherwise standard notion of unforgeability [PS14,PS17], where adversaries can adaptively query signatures for messages but only on distinct addresses. Figure 2 details the unforgeability security experiment.

Definition 5 ($\mathsf{EUF\text{-}CMA}$ [PS14]). *For a PPT adversary* $\mathcal{A}$*, we define the advantage function in the sense of* $\mathsf{EUF\text{-}CMA}$ *as*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = \Pr\left[\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = 1\right]$$

where the corresponding experiment is depicted in Figure 2. If for all PPT adversaries $\mathcal{A}$ *there is a negligible function* $\varepsilon(\cdot)$ *such that*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) \leq \varepsilon(\kappa)$$

we say that DAPS *is* $\mathsf{EUF\text{-}CMA}$ *secure.*

The interesting property of a DAPS scheme is the notion of double-signature extractability (DSE). It requires that whenever one obtains signatures on two colliding messages, one should be able to extract the signing key using the extraction algorithm $\mathsf{Ex_D}$. We present the security definition denoted as DSE in Figure 3. Thereby, we consider the common notion which requires extraction to work if the key pair has been generated honestly. In this game, the adversary is given a key pair and outputs two colliding messages and corresponding signatures. The adversary wins the game if the key produced by $\mathsf{Ex_D}$ is different from the signing key, although extraction should have succeeded, i.e, the messages were colliding and their signatures were valid.

```
$\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\mathsf{DAPS}}(\kappa)$:
  $(\mathsf{sk_D}, \mathsf{pk_D}) \leftarrow \mathsf{KGen_D}(1^\kappa)$
  $\mathcal{Q} \leftarrow \emptyset$, $\mathcal{R} \leftarrow \emptyset$
  $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}'_\mathsf{D}(\mathsf{sk_D},\cdot)}(\mathsf{pk}_\Sigma)$
    where oracle $\mathsf{Sign}'_\mathsf{D}$ on input $m$:
      $(a, p) \leftarrow m$
      if $a \in \mathcal{R}$, return $\perp$
      $\sigma \leftarrow \mathsf{Sign_D}(\mathsf{sk_D}, m)$, $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$, $\mathcal{R} \leftarrow \mathcal{R} \cup \{a\}$
      return $\sigma$
  return 1, if $\mathsf{Verify_D}(\mathsf{pk_D}, m^*, \sigma^*) = 1 \ \wedge \ m^* \notin \mathcal{Q}$
  return 0
```

Fig. 2: EUF-CMA **security for** DAPS**.**

Definition 6 (DSE [**PS14**])**.** *For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of double-signature extraction (*DSE*) as*

$$\mathsf{Adv}^{\mathsf{DSE}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = \Pr\left[\mathsf{Exp}^{\mathsf{DSE}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = 1\right]$$

where the corresponding experiment is depicted in Figure 3. If for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{DSE}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) \leq \varepsilon(\kappa),$$

then DAPS *provides* DSE*.*

```
$\mathsf{Exp}^{\mathsf{DSE}}_{\mathcal{A},\mathsf{DAPS}}(\kappa)$:
  $(\mathsf{sk_D}, \mathsf{pk_D}) \leftarrow \mathsf{KGen_D}(1^\kappa)$
  $(m_1, m_2, \sigma_1, \sigma_2) \leftarrow \mathcal{A}(\mathsf{sk_D}, \mathsf{pk_D})$
  return 0, if $m_1$ and $m_2$ are not colliding
  return 0, if $\mathsf{Verify_D}(\mathsf{pk_D}, m_i, \sigma_i) = 0$ for any $i \in [2]$
  $\mathsf{sk}'_\mathsf{D} \leftarrow \mathsf{Ex_D}(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2)$
  return 1, if $\mathsf{sk}'_\mathsf{D} \neq \mathsf{sk_D}$
  return 0
```

Fig. 3: DSE **security for** DAPS**.**

In Appendix A we recall the strong variant of extractability under malicious keys (denoted as DSE^*), where the adversary is allowed to generate the key arbitrarily. The DSE^* notion is very interesting from a theoretical perspective, but no practically efficient DAPS construction can achieve this notion so far.

DRS in [DRS18b] argue that when DAPS are constructed by extending a conventional signature scheme Σ, extraction of the part of the signing key corresponding to Σ is already sufficient to disincentivizes double-authentication for many applications. Hence, Derler et al. [DRS18b] defined two weaker double-signature extraction notions that cover extraction of the signing key of the underlying signature scheme for honestly and maliciously generated DAPS keys. The security games for weak double-signature extraction (wDSE) and weak double-signature extraction under malicious keys (wDSE*) are depicted in Figure 4 and Figure 5. DSE and DSE* imply their weaker counterparts and wDSE* implies wDSE.

Definition 7 **($T \in \{\mathsf{wDSE}, \mathsf{wDSE}^*\}$).** *For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of weak double-signature extraction ($T = \mathsf{wDSE}$) and weak double-signature extraction under malicious keys ($T = \mathsf{wDSE}^*$), as*

$$\mathsf{Adv}^{T}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = \Pr\left[\mathsf{Exp}^{T}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = 1\right]$$

where the corresponding experiments are depicted in Figure 4 and Figure 5 respectively. If for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{T}_{\mathcal{A},\mathsf{DAPS}}(\kappa) \leq \varepsilon(\kappa),$$

then DAPS *provides T.*

```
$\mathsf{Exp}^{\mathsf{wDSE}}_{\mathcal{A},\mathsf{DAPS}}(\kappa)$:
  $(\mathsf{sk_D}, \mathsf{pk_D}) \leftarrow \mathsf{KGen_D}(1^\kappa)$ with $\mathsf{sk_D} = (\mathsf{sk}_\Sigma, \dots)$
  $(m_1, m_2, \sigma_1, \sigma_2) \leftarrow \mathcal{A}(\mathsf{sk_D}, \mathsf{pk_D})$
  return 0, if $m_1$ and $m_2$ are not colliding
  return 0, if $\mathsf{Verify_D}(\mathsf{pk_D}, m_i, \sigma_i) = 0$ for any $i \in [2]$
  $\mathsf{sk}'_\mathsf{D} \leftarrow \mathsf{Ex_D}(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2)$ where $\mathsf{sk}'_\mathsf{D} = (\mathsf{sk}'_\Sigma, \dots)$
  return 1, if $\mathsf{sk}'_\Sigma \neq \mathsf{sk}_\Sigma$
  return 0
```

Fig. 4: wDSE **security for** DAPS.

```
$\mathsf{Exp}^{\mathsf{wDSE}^*}_{\mathcal{A},\mathsf{DAPS}}(\kappa)$:
  $(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2) \leftarrow \mathcal{A}(1^\kappa)$ where $\mathsf{pk_D} = (\mathsf{pk}_\Sigma, \dots)$
  return 0, if $m_1$ and $m_2$ are not colliding
  return 0, if $\mathsf{Verify_D}(\mathsf{pk_D}, m_i, \sigma_i) = 0$ for any $i \in [2]$
  $\mathsf{sk}'_\mathsf{D} \leftarrow \mathsf{Ex_D}(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2)$ where $\mathsf{sk}'_\mathsf{D} = (\mathsf{sk}'_\Sigma, \dots)$
  return 1, if $\mathsf{sk}'_\Sigma$ is not the secret key corresponding to $\mathsf{pk}_\Sigma$
  return 0
```

Fig. 5: wDSE* **security for** DAPS.

Finally, for our constructions we may sometimes require a very mild additional property of DAPS which we call *verifiability of secret keys*. Informally it requires that there is an additional efficient algorithm VKey which, given a key pair, outputs 1 if the given secret key is the key corresponding to the given public key. Formally we define verifiability of keys as follows:

Definition 8 (Verifiability of Keys). *We say that a DAPS scheme* $\mathsf{DAPS} = (\mathsf{KGen}_\mathsf{D}, \mathsf{Sign}_\mathsf{D}, \mathsf{Verify}_\mathsf{D}, \mathsf{Ex}_\mathsf{D})$ *provides verifiability of keys, if it provides an additional efficient algorithm* VKey *so that for all* $\kappa \in \mathbb{N}$*, for all* $(\mathsf{sk}, \mathsf{pk})$ *it holds that*

$$\mathsf{VKey}(\mathsf{sk}, \mathsf{pk}) = 1 \implies (\mathsf{sk}, \mathsf{pk}) \in \mathsf{KGen}_\mathsf{D}(1^\kappa).$$

2.3 Non-Interactive ZK Proof Systems (NIZK)

We recall a standard definition of non-interactive zero-knowledge proof systems. Let $L \subseteq \mathsf{X}$ be an **NP**-language with associated witness relation R so that $L = \{x \mid \exists w : R(x, w) = 1\}$.

Definition 9 (Non-Interactive Zero-Knowledge Proof System). *A non-interactive proof system* Π *is a tuple of algorithms* $(\mathsf{Setup}_\Pi, \mathsf{Proof}_\Pi, \mathsf{Verify}_\Pi)$*, which are defined as follows:*

$\mathsf{Setup}_\Pi(1^\kappa)$: *This algorithm takes a security parameter* κ *as input, and outputs a common reference string* crs.

$\mathsf{Proof}_\Pi(\mathsf{crs}, x, w)$: *This algorithm takes a common reference string* crs*, a statement* x*, and a witness* w *as input, and outputs a proof* π*.*

$\mathsf{Verify}_\Pi(\mathsf{crs}, x, \pi)$: *This algorithm takes a common reference string* crs*, a statement* x*, and a proof* π *as input, and outputs a bit* $b \in \{0, 1\}$*.*

From a non-interactive zero-knowledge proof system we require *completeness*, *soundness* and *adaptive zero-knowledge* and *simulation-sound extractability*. In Appendix C we recall formal definitions of those properties.

NIZK from Σ-protocols. A Σ-protocol for language L is an interactive three move protocol between a prover and a verifier, where the prover proves knowledge of a witness w to the statement $x \in L$. We recall the formal definition of Σ-protocols in Appendix B. One can obtain a non-interactive proof system with the above properties by applying the Fiat-Shamir transform [FS86] to any Σ-protocol where the min-entropy μ of the commitment a sent in the first message of the Σ-protocol is so that $2^{-\mu}$ is negligible in the security parameter κ and its challenge space C is exponentially large in the security parameter. Essentially, the transform removes the interaction between the prover and the verifier by using a hash function H (modelled as a random oracle) to obtain the challenge. That is, the algorithm $\mathsf{Challenge}$ obtains the challenge as $H(\mathsf{a}, x)$. Due to the lack of space we postpone a formal presentation to Appendix C.1.

Efficient NIZK Proof Systems for General Circuits. Over the last few years NIZK proof systems for general circuits have seen significant progress improving their overall efficiency. Based on the MPC-in-the-head paradigm by Ishai et al. [IKOS09], ZKBoo [GMO16] and the optimized version ZKB++ [CDG+17a] are zero-knowledge proof systems covering languages over arbitrary circuits. They roughly work as follows: The prover simulates all parties of a multiparty computation (MPC) protocol implementing the joint evaluation of some function, say $y = \text{SHA-3}(x)$, and computes commitments to the states of all players. The verifier then randomly corrupts a subset of the players and checks whether those players performed the computation correctly. Following the same paradigm, Katz et al. [KKW18] recently proposed to use a MPC protocol with a preprocessing phase, which allows to significantly reduce the proof sizes. This proof system, denoted as KKW, allows one to choose a larger number of players then in the case of ZKBoo and ZKB++, where larger numbers lead to smaller proofs. For all three proof systems, the number of binary multiplication gates is the main factor influencing the proof size, as the proof size grows linearly with the number of those gates.

Finally, Ames et al. [AHIV17] introduced Ligero, which offers proofs of logarithmic size in the number of multiplication gates if the circuit is represented using a prime field. When considering binary circuits, the number of addition respectively XOR gates has also to be accounted for in the proof size. But, as noted by Katz et al. in [KKW18], especially for large circuits with more than 100,000 gates Ligero beats ZKBoo, ZKB++ and KKW in term of proof size.

2.4 Shamir's Secret Sharing

Shamir's (k, ℓ)-threshold secret sharing [Sha79] is a secret sharing scheme which allows to information-theoretically share a secret s among a set of ℓ parties so that any collection of at least k shares allow to reconstruct s. Let s be the constant term of an otherwise randomly chosen $k - 1$ degree polynomial

$$f(X) = \rho_{k-1}X^{k-1} + \cdots + \rho_1 X + s$$

over a finite field $\mathbb{F}$. A share is computed as $f(i)$ for party i, $1 \leq i \leq \ell$. Let $\mathcal{S}$ be any set of cardinality at least k of these ℓ shares and let $I_{\mathcal{S}}$ be the set of indices corresponding to shares in $\mathcal{S}$. Using Lagrange interpolation one can then can reconstruct the secret s by computing $s = f(0)$ as

$$s = \sum_{j \in I_{\mathcal{S}}} \lambda_j f(j) \quad \text{with} \quad \lambda_j = \prod_{i \in I_{\mathcal{S}} \setminus \{j\}} \frac{j}{j - i}.$$

As long as only $k - 1$ or less shares are available the secret s is information-theoretically hidden.

3 DAPS without Structured Hardness Assumptions

For our first construction we follow the basic idea of Derler et al. [DRS18b] and build DAPS by including secret shares of the signing key in the signatures. To

```
$\mathsf{KGen}_\mathsf{D}(1^\kappa)$: Fix a signature scheme $\Sigma = (\mathsf{KGen}_\Sigma, \mathsf{Sign}_\Sigma, \mathsf{Verify}_\Sigma)$, a value-key-binding
    PRF $\mathcal{F} : \mathcal{S} \times D \to \mathsf{R}$ with respect to $\beta \in D$. Let $\mathsf{sk}_\mathsf{PRF} \overset{R}{\leftarrow} \mathcal{S}$, and $\mathsf{crs} \leftarrow \mathsf{Setup}_\Pi(1^\kappa)$.
    Let $c = \mathcal{F}(\mathsf{sk}_\mathsf{PRF}, \beta)$. Set $\mathsf{sk}_\mathsf{D} \leftarrow (\mathsf{sk}_\Sigma, \mathsf{sk}_\mathsf{PRF})$, $\mathsf{pk}_\mathsf{D} \leftarrow (\mathsf{pk}_\Sigma, \mathsf{crs}, \beta, c)$.
$\mathsf{Sign}_\mathsf{D}(\mathsf{sk}_\mathsf{D}, m)$: Parse $\mathsf{sk}_\mathsf{D}$ as $(\mathsf{sk}_\Sigma, \mathsf{sk}_\mathsf{PRF})$ and $m$ as $(a, p)$.
    1. $\rho \leftarrow \mathcal{F}(\mathsf{sk}_\mathsf{PRF}, a)$
    2. $z \leftarrow \rho p + \mathsf{sk}_\Sigma$
    3. $\pi \leftarrow \mathsf{Proof}_\Pi(\mathsf{crs}, (\mathsf{pk}_\Sigma, \beta, c, a, z, m), (\mathsf{sk}_\Sigma, \mathsf{sk}_\mathsf{PRF}, \rho))$
    4. Return $(z, \pi)$.
$\mathsf{Verify}_\mathsf{D}(\mathsf{pk}_\mathsf{D}, m, \sigma)$: Parse $\mathsf{pk}_\mathsf{D}$ as $(\mathsf{pk}_\Sigma, \mathsf{crs}, \beta, c)$, $m$ as $(a, p)$ and $\sigma$ as $(z, \pi)$.
    1. Return $\mathsf{Verify}_\Pi(\mathsf{crs}, (\mathsf{pk}_\Sigma, \beta, c, a, z, m), \pi)$.
$\mathsf{Ex}_\mathsf{D}(\mathsf{pk}_\mathsf{D}, m_1, m_2, \sigma_1, \sigma_2)$: Parse $\sigma_i$ as $(z_i, \cdot)$, $m_i$ as $(a_i, p_i)$.
    1. If $m_1$ and $m_2$ are not colliding, return $\perp$
    2. if $\mathsf{Verify}_\mathsf{D}(\mathsf{pk}_\mathsf{D}, m_i, \sigma_i) = 0$ for any $i$, return $\perp$
    3. let $\mathsf{sk}_\Sigma \leftarrow \frac{z_1 p_2 - z_2 p_1}{p_2 - p_1}$
    4. return $\mathsf{sk}_\Sigma$
```

Scheme 1: Generic DAPS from Σ.

resolve the address space limitation of their approach, however, we derive the coefficients of the sharing polynomial using a pseudorandom function (PRF). By then additionally proving the correct evaluation of the PRF, it is no longer necessary to store encrypted versions of the coefficients in the public key. The only issue which remains, is to additionally prove consistency with respect to a "commitment" to the PRF secret key contained in the public key (we commit to it using a fixed-value key-binding PRF as defined in Appendix D). To bind the message to the proof, we use a signature-of-knowledge style methodology [CL06].

More precisely, we start from a one-way function $f : S \to P$, which we use to define the relation between public and secret keys, i.e., so that $\mathsf{pk}_\Sigma = f(\mathsf{sk}_\Sigma)$. In addition we carefully choose a PRF $\mathcal{F}$, which maps to the secret key space S. At the core of our DAPS construction we use a NIZK proof to prove consistency of the secret signing key, as well as the correctness of the secret sharing. For this proof we define an language L with associated witness relation R in the following way:

$$((\mathsf{pk}_\Sigma, \beta, c, a, z), (\mathsf{sk}_\Sigma, \mathsf{sk}_\mathsf{PRF}, \rho)) \in R \iff$$
$$\rho = \mathcal{F}(\mathsf{sk}_\mathsf{PRF}, a) \ \wedge \ z = \rho p + \mathsf{sk}_\Sigma \ \wedge \ c = \mathcal{F}(\mathsf{sk}_\mathsf{PRF}, \beta) \ \wedge \ \mathsf{pk}_\Sigma = f(\mathsf{sk}_\Sigma)$$

In this statement we cover three aspects: First, we prove that the polynomial for Shamir's secret sharing is derived from the address and that the secret share is correctly calculated. Second, we prove the relation between the secret and public key of the signature scheme. Third, we "commit" to the PRF secret key using a fixed-value key-binding PRF. The full scheme is depicted in Scheme 1.

It is important to note that the PRF needs to be compatible with the signature scheme, in the sense that secret-key space of Σ, i.e., S, and R match. For simplicity, we assume that $\mathsf{R} = S$. Additionally, the domain and codomain of the PRF also define the message space of the DAPS. In the following theorem we prove that Scheme 1 is an $\mathsf{EUF\text{-}CMA}$-secure DAPS.

Theorem 1. *If the NIZK proof system* Π *is simulation-sound extractable,* $\mathcal{F}$ *is a PRF, and* f *is an OWF, then Scheme 1 provides* $\mathsf{EUF\text{-}CMA}$ *security.*

Proof. We prove this theorem using a sequence of games. We denote the winning event of game G_i as S_i. We let Q_Σ be the number of signing oracle queries.

Game 0: The original game.

Game 1: As before, but we modify KGen_D as follows:
$\mathsf{KGen}_\mathsf{D}(1^\kappa)\colon$ As before, but let $\boxed{(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_{1,\Pi}(1^\kappa)}$ and store $\boxed{\tau}$.

Transition $0 \Rightarrow 1$**:** Both games are indistinguishable under adaptive zero-knowledge of the proof system, i.e. $|\Pr[S_0] - \Pr[S_1]| \le \mathsf{Adv}^{\mathsf{Sim}}_{\mathcal{A},\mathsf{S},\Pi}(\kappa)$.

Game 2: As Game 1, but we modify Sign_D as follows:
$\mathsf{Sign}_\mathsf{D}(\mathsf{sk}, m)\colon$ As before, but let $\boxed{\pi \leftarrow \mathcal{S}_{2,\Pi}(\mathsf{crs}, \tau, (\mathsf{pk}_\Sigma, \beta, c, a, z, m))}$.

Transition $1 \Rightarrow 2$**:** Both games are indistinguishable under adaptive zero-knowledge of the proof system, i.e. $|\Pr[S_1] - \Pr[S_2]| \le \mathsf{Adv}^{\mathsf{ZK}}_{\mathcal{A},\mathsf{S},\Pi}(\kappa)$.

Game 3: As before, but we modify KGen_D and Sign_D as follows.
$\mathsf{KGen}_\mathsf{D}(1^\kappa)\colon$ As before, but let $\boxed{c \xleftarrow{R} \mathsf{R}}$.
$\mathsf{Sign}_\mathsf{D}(\mathsf{sk}_\mathsf{D}, m)\colon$ As before, but let $\boxed{\rho \xleftarrow{R} \mathsf{R}}$.

Transition $2 \Rightarrow 3$**:** We engage with a PRF challenger $\mathcal{C}$ against $\mathcal{F}$. We modify Sign_D as follows:
$\mathsf{KGen}_\mathsf{D}(1^\kappa)\colon$ As before, but let $\boxed{c \xleftarrow{R} \mathcal{C}(\beta)}$.
$\mathsf{Sign}_\mathsf{D}(\mathsf{sk}_\mathsf{D}, m)\colon$ As before, but let $\boxed{\rho \xleftarrow{R} \mathcal{C}(a)}$.
Thus an adversary distinguishing the two games also distinguishes the PRF from a random function, i.e. $|\Pr[S_4] - \Pr[S_3]| \le \mathsf{Adv}_{\mathcal{D},F}(\kappa)$.

Game 4: As before, but we modify Sign_D as follows.
$\mathsf{Sign}_\mathsf{D}(\mathsf{sk}_\mathsf{D}, m)\colon$ As before, but track all (a, ρ) pairs in $\mathcal{Q}$.
We abort if there exists $(a_1, \rho), (a_2, \rho) \in \mathcal{Q}$ such that $a_1 \neq a_2$.

Transition $3 \Rightarrow 4$**:** Both games proceed identically, unless the abort event happens. The probability of the abort event is bounded by $1/|\mathsf{R}|$, i.e. $|\Pr[S_5] - \Pr[S_4]| \le Q_\Sigma/|\mathsf{R}|$.

Game 5: As before, but we modify Sign_D as follows.
$\mathsf{Sign}_\mathsf{D}(\mathsf{sk}_\mathsf{D}, m)\colon$ As before, but let $\boxed{z \xleftarrow{R} \mathsf{R}}$.

Transition $4 \Rightarrow 5$**:** This change is conceptional. Note that ρ is uniformly random and not revealed, and thus z is uniformly random.

Game 6: As before, but we modify KGen_D as follows:
$\mathsf{KGen}_\mathsf{D}(1^\kappa)\colon$ As before, but let $\boxed{(\mathsf{crs}, \tau, \xi) \leftarrow \mathcal{E}_{1,\Pi}(1^\kappa)}$ and store $\boxed{(\tau, \xi)}$.

Transition $5 \Rightarrow 6$**:** Both games are indistinguishable under simulation-sound extractability of the proof system, i.e. $|\Pr[S_6] - \Pr[S_5]| \le \mathsf{Adv}^{\mathsf{Ext}_1}_{\mathcal{A},\mathcal{E},\Pi}(\kappa)$.

Game 7: As before, but we now use the extractor to obtain $\mathsf{sk}^*_\Sigma \leftarrow \mathcal{E}_{2,\Pi}(\mathsf{crs}, \xi, (\mathsf{pk}_\Sigma, \beta, c, a, z, m), \pi)$ and abort in case the extraction fails.

Transition $6 \Rightarrow 7$**:** Both games proceed identically, unless we abort. The probability of that happening is bounded by the simulation-sound extractablity of the proof system, i.e. $|\Pr[S_7] - \Pr[S_6]| \le \mathsf{Adv}^{\mathsf{Ext}_2}_{\mathcal{A},\mathcal{E},\Pi}(\kappa)$.

Reduction. Now we are ready to present a reduction which engages with an OWF challenger $\mathcal{C}$. In particular, we obtain a challenge and embed it in the public key, i.e.

$\mathsf{KGen}_\mathsf{D}(1^\kappa)$: As before, but $\boxed{\mathsf{pk}_\Sigma \leftarrow \mathcal{C}}$.

Once the adversary returns a forgery, we extract sk_Σ^* and forward the solution to the OWF challenger. Hence $\Pr[S_7] \leq \mathsf{Adv}_{\mathcal{A},f}^{\mathsf{OWF}}(\kappa)$, which concludes the proof. □

We now show that Scheme 1 also provides wDSE security. We note that in the proof of Theorem 2 we do not need to simulate proofs, so a weaker extraction notion would suffice. The proof of Theorem 1, however, already requires simulation-sound extractability which is why we directly resort to simulation-sound extractability.

Theorem 2. *If the NIZK proof system* Π *is simulation-sound extractable and the PRF* $\mathcal{F}$ *is computationally fixed-value-key-binding, then Scheme 1 provides* wDSE *security.*

Proof. We prove this theorem using a sequence of games. We denote the winning event of game G_i as S_i. Let $m_1, m_2, \sigma_1, \sigma_2$ denote the output of $\mathcal{A}$. For simplicity we write $m_j = (a, p_j)$, $\sigma_j = (z_j, \pi_j)$ for $j \in [2]$. Now, we have proofs attesting that $z_j = \rho p_j + \mathsf{sk}_\Sigma$ for $j \in [2]$.

Game 0: The original game.
Game 1: As before, but we modify KGen_D as follows:
$\mathsf{KGen}_\mathsf{D}(1^\kappa)$: As before, but let $\boxed{(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_{1,\Pi}(1^\kappa)}$ and store $\boxed{\tau}$.
Transition $0 \Rightarrow 1$**:** Both games are indistinguishable under adaptive zero-knowledge of the proof system, i.e. $|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{Adv}_{\mathcal{A},\mathsf{S},\Pi}^{\mathsf{Sim}}(\kappa)$.
Game 2: As before, but we modify KGen_D as follows:
$\mathsf{KGen}_\mathsf{D}(1^\kappa)$: As before, but let $\boxed{(\mathsf{crs}, \tau, \xi) \leftarrow \mathcal{E}_{1,\Pi}(1^\kappa)}$ and store $\boxed{\xi}$.
Transition $1 \Rightarrow 2$**:** Both games are indistinguishable under simulation-sound extractability of the proof system, i.e. $|\Pr[S_2] - \Pr[S_1]| \leq \mathsf{Adv}_{\mathcal{A},\mathcal{E},\Pi}^{\mathsf{Ext}_1}(\kappa)$.
Game 3: As before, but we now use the extractor to obtain $(\mathsf{sk}_{\Sigma,j}^*, \mathsf{sk}_{\mathsf{PRF},j}^*) \leftarrow \mathcal{E}_{2,\Pi}(\mathsf{crs}, \xi, (\mathsf{pk}_\Sigma, \beta, c, a, z_j, m_j), \pi)$ for $j \in [2]$ and abort if the extraction fails.
Transition $2 \Rightarrow 3$**:** Both games proceed identically, unless we abort. The probability of that happening is bounded by the simulation-sound extractablity of the proof system, i.e. $|\Pr[S_3] - \Pr[S_2]| \leq 2 \cdot \mathsf{Adv}_{\mathcal{A},\mathcal{E},\Pi}^{\mathsf{Ext}_2}(\kappa)$.
Game 4: As before, but we abort if $\mathsf{sk}_\mathsf{PRF} \neq \mathsf{sk}_{\mathsf{PRF},j}^*$ for any $j \in [2]$.
Transition $3 \Rightarrow 4$**:** Both games proceed identically, unless we abort. Let $j \in [2]$ be such that $\mathsf{sk}_\mathsf{PRF} \neq \mathsf{sk}_{\mathsf{PRF},j}^*$. We bound the abort probability using $\mathcal{F}$. Let $\mathcal{C}$ be a computational fixed-value-key-binding challenger. We modify KGen_D as follows:
$\mathsf{KGen}_\mathsf{D}(1^\kappa)$: As before, but let $\boxed{(\mathsf{sk}_\mathsf{PRF}, \beta)} \leftarrow \mathcal{C}$.
Then we have that $\mathcal{F}(\mathsf{sk}_\mathsf{PRF}, \beta) = \mathcal{F}(\mathsf{sk}_{\mathsf{PRF},j}^*, \beta)$, hence we forward $\mathsf{sk}_{\mathsf{PRF},j}^*$ to $\mathcal{C}$. Thus we built an adversary $\mathcal{B}$ against fixed-value-key-binding of $\mathcal{F}$, i.e. $|\Pr[S_4] - \Pr[S_3]| \leq \mathsf{Adv}_{\mathcal{B},\mathcal{F}}^{\mathsf{cFKVB}}(\kappa) = \varepsilon(\kappa)$.

As we have now ensured that the correct PRF secret key was used to generate ρ from a, sk_Σ is now uniquely determined via the secret sharing. Thus the adversary can no longer win, i.e. $\Pr[S_4] = 0$. □

Extension to NAPS. Following the ideas outlined in [DRS18b], Scheme 1 can be extended to an N-time authentication-preventing signature scheme by changing the sharing polynomial $\rho X + \mathsf{sk}_\Sigma$ to a polynomial of degree $N-1$ with coefficients $\rho_1, \ldots, \rho_{N-1}$ obtained from the PRF via $\rho_i = \mathcal{F}(\mathsf{sk}_{\mathsf{PRF}}, a \| i)$.

Instantiations. The requirement on the signature scheme are very weak, yet finding a suitable combination of primitives can be difficult. Thus we discuss some possible instantiations. One candidate scheme on top of which the DAPS extension can be applied is Picnic [CDG+17a,CDG+17b]. In Picnic the public key pk_Σ is the image of the secret key sk_Σ under a one-way function built from LowMC [ARS+15,ARS+16]. Signatures are then generated by proving this relation using a NIZK from ZKB++ made non-interactive. In this case it is straight forward to use the block cipher LowMC (denoted by $\mathcal{E}$) as PRF by setting $\mathcal{F}(s, x) = \mathcal{E}(s, x) \oplus x$. We argue that this PRF can also be considered a computational fixed-value-key-binding PRF, since it is reasonable to assume that finding a new key which maps one particular input to one particular output is no easier than generic key search. Furthermore, when increasing the block size of LowMC relative to the key size, the existence of second key mapping to the same output becomes increasingly unlikely.

The circuit for the secret sharing can either be implemented using a binary circuit realizing the required arithmetic, or, more efficiently, by computing the sharing bit-wise. For the latter, we consider ρ, p and sk_Σ as n bit values, and compute secret shares $z_i = \rho_i p_i + \mathsf{sk}_{\Sigma,i}$ for each bit $i \in [n]$. Thus only n ANDs are required to implemented the secret sharing. All in all Picnic signatures can be easily extended to a DAPS without requiring extensive changes. We also note that the Fiat-Shamir transformed ZKB++ is in fact simulation-sound extractable NIZK proof systems as confirmed in [DRS18a]. Using the signature size formulas, we can estimate DAPS signatures sizes at around 408 KB, meaning there is a overhead of 293 KB compared to Picnic signatures requiring roughly 115 KB in the ROM targeting 256 bit classical security. Analogously to the QROM security of Picnic, Unruh's transform [Unr12,Unr15,Unr16] can be used to obtain QROM security for the DAPS construction.

Also hash-based signatures such as SPHINCS [BHH+15] are well suited for this construction. Similar to the case of Picnic, the PRF can be instantiated using LowMC. However, the consistency proof is more expensive, as computing the public key requires multiple evaluations of hash functions.

Relying on Structured Hardness Assumptions. The situation is different for signature schemes relying on structured hardness assumptions, e.g., those in the discrete logarithm setting such as Schnorr signatures [Sch89], ECDSA and EdDSA [BDL+12]. While they would fulfill the requirement for the secret-key-to-public-key relation, i.e., here working in a group $\mathbb{G}$ with generator g the OWF is of the form $f(x) := g^x$, the problem is finding an efficient NIZK proof system to prove statements over $\mathbb{Z}_p$ and in a prime order group $\mathbb{G}$ simultaneously. Furthermore the NIZK proof system would also need to support statements over binary circuits for the PRF evaluation. Recently, Agrawal et al. [AGM18] made progress in this direction, enabling non-interactive proofs of composite statements for relations over multiple groups and binary circuits. Using these techniques to construct DAPS is an interesting open problem.

4 Extending Any Signature Scheme Using DAPS

Finally, we follow a different direction for our second approach. Here we start from an already existing DAPS and use it to extend *any* unforgeable signature scheme to a DAPS. Interestingly, both the unforgeability and extraction follow in a black-box way from the signature scheme and the underlying DAPS, respectively. In this construction, the secret key consists of the secret keys of the underlying DAPS and signature scheme. To guarantee extraction of the full secret key, we apply the technique of Bellare et al. [BPS17] and encrypt the key of the signature scheme using a one-time pad derived from the secret key of the DAPS scheme. The public key then consists of that encrypted key and the public keys of the underlying DAPS and signature scheme. However, for extraction of maliciously generated keys, i.e., DSE*-security, this means that public keys need to be extended with a NIZK proof that the encryption was performed correctly. For the sake of simplicity, we thus concentrate on the DSE security of the scheme. We present the compiler in Scheme 2.

```
$\mathsf{KGen_D}(1^\kappa)$: Fix some signature scheme $\Sigma = (\mathsf{KGen}_\Sigma, \mathsf{Sign}_\Sigma, \mathsf{Verify}_\Sigma)$ and some DAPS
    $\mathsf{DAPS} = (\mathsf{KGen_D}, \mathsf{Sign_D}, \mathsf{Verify_D}, \mathsf{Ex_D})$ with verifiability of keys. Let $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow$
    $\Sigma.\mathsf{KGen}_\Sigma(1^\kappa)$, $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{DAPS}.\mathsf{KGen_D}(1^\kappa)$, $Y \leftarrow \mathsf{sk}_\Sigma \oplus H(\mathsf{sk})$, and return
    $(\mathsf{sk_D}, \mathsf{pk_D}) := ((\mathsf{sk}_\Sigma, \mathsf{sk}), (\mathsf{pk}_\Sigma, \mathsf{pk}, Y))$.
$\mathsf{Sign_D}(\mathsf{sk_D}, m)$: Parse $\mathsf{sk_D}$ as $(\mathsf{sk}_\Sigma, \mathsf{sk})$.
    1. $\sigma_0 \leftarrow \Sigma.\mathsf{Sign}_\Sigma(\mathsf{sk}_\Sigma, m)$
    2. $\sigma_1 \leftarrow \mathsf{DAPS}.\mathsf{Sign_D}(\mathsf{sk}, m)$
    3. Return $\sigma = (\sigma_0, \sigma_1)$
$\mathsf{Verify_D}(\mathsf{pk_D}, m, \sigma)$: Parse $\mathsf{pk_D}$ as $(\mathsf{pk}_\Sigma, \mathsf{pk}, \cdot)$, and return 1 if all of the following checks
    hold and 0 otherwise:
    – $\Sigma.\mathsf{Verify}_\Sigma(\mathsf{pk}, (a, p)) = 1$
    – $\mathsf{DAPS}.\mathsf{Verify_D}(\mathsf{pk_D}, (a, p)) = 1$
$\mathsf{Ex_D}(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2)$: Parse $\mathsf{pk_D}$ as $(\mathsf{pk}_\Sigma, \mathsf{pk}, Y)$, obtain $\mathsf{sk} \leftarrow \mathsf{DAPS}.\mathsf{Ex_D}(\mathsf{pk}, m_1, m_2,$
    $\sigma_1, \sigma_2)$ and $\mathsf{sk}_\Sigma \leftarrow Y \oplus H(\mathsf{sk})$, and return $\mathsf{sk_D} = (\mathsf{sk}_\Sigma, \mathsf{sk})$.
```

Scheme 2: Black-Box Extension of any Signature Scheme to DAPS.

In the following theorem we formally state that the DAPS construction in Scheme 2 yields an EUF-CMA-secure DAPS.

Theorem 3. *If Σ is unforgeable,* DAPS *is unforgeable and provides verifiability of keys, then the DAPS construction in Scheme 2 is unforgeable in the ROM.*

The theorem above is proven in Appendix E.1. Additionally, Scheme 2 provides DSE-security if the underlying DAPS provides it as well.

Theorem 4. *If* DAPS *provides* DSE*-security, then the construction of DAPS in Scheme 2 provides* DSE*-security as well.*

The theorem above is proven in Appendix E.2.

5 Conclusion

In this work, we close two important gaps in the literature on DAPS. First, we present a generic DAPS construction, which, in contrast to [DRS18b], does not come with the drawback of a polynomially bounded address space. Our construction only relies on assumptions related to symmetric key primitives, which is why we also obtain a candidate for a post-quantum DAPS construction. Second, we also present an alternative generic construction of DAPS which basically shows how to bring DAPS features to any signature scheme. This is of particular practical importance, as it allows to extend arbitrary signature schemes with double signature extraction features. As our compiler works by using an arbitrary DAPS scheme to extend a given signature scheme in a black-box way, this yields more efficient DAPS than previously known for standardized and widely used signature schemes such as ECDSA or EdDSA.

References

[AGM18] Shashank Agrawal, Chaya Ganesh, and Payman Mohassel. Non-interactive zero-knowledge proofs for composite statements. In *CRYPTO (3)*, volume 10993 of *Lecture Notes in Computer Science*, pages 643–673. Springer, 2018.

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In *ACM Conference on Computer and Communications Security*, pages 2087–2104. ACM, 2017.

[ARS$^+$15] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In *EUROCRYPT (1)*, volume 9056 of *Lecture Notes in Computer Science*, pages 430–454. Springer, 2015.

[ARS$^+$16] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. *IACR Cryptology ePrint Archive*, 2016:687, 2016.

[BDL$^+$12] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. *J. Cryptographic Engineering*, 2(2):77–89, 2012.

[BEF18] Dan Boneh, Saba Eskandarian, and Ben Fisch. Post-quantum group signatures from symmetric primitives. *IACR Cryptology ePrint Archive*, 2018:261, 2018.

[BHH$^+$15] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. SPHINCS: practical stateless hash-based signatures. In *EUROCRYPT (1)*, volume 9056 of *Lecture Notes in Computer Science*, pages 368–397. Springer, 2015.

[BKN17] Dan Boneh, Sam Kim, and Valeria Nikolaenko. Lattice-based DAPS and generalizations: Self-enforcement in signature schemes. In *ACNS*, volume 10355 of *Lecture Notes in Computer Science*, pages 457–477. Springer, 2017.

[BPS16] Mihir Bellare, Bertram Poettering, and Douglas Stebila. From identification to signatures, tightly: A framework and generic transforms. In *ASIACRYPT (2)*, volume 10032 of *Lecture Notes in Computer Science*, pages 435–464, 2016.

[BPS17] Mihir Bellare, Bertram Poettering, and Douglas Stebila. Deterring certificate subversion: Efficient double-authentication-preventing signatures. In *Public Key Cryptography (2)*, volume 10175 of *Lecture Notes in Computer Science*, pages 121–151. Springer, 2017.

[CDG$^+$17a] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In *ACM Conference on Computer and Communications Security*, pages 1825–1842. ACM, 2017.

[CDG$^+$17b] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. The Picnic Signature Algorithm Specification, 2017. https://github.com/Microsoft/Picnic/blob/master/spec.pdf.

[CL06] Melissa Chase and Anna Lysyanskaya. On signatures of knowledge. In *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 78–96. Springer, 2006.

[CMR98] Ran Canetti, Daniele Micciancio, and Omer Reingold. Perfectly one-way probabilistic hash functions (preliminary version). In *STOC*, pages 131–140. ACM, 1998.

[DRS18a] David Derler, Sebastian Ramacher, and Daniel Slamanig. Post-quantum zero-knowledge proofs for accumulators with applications to ring signatures from symmetric-key primitives. In *PQCrypto*, volume 10786 of *Lecture Notes in Computer Science*, pages 419–440. Springer, 2018.

[DRS18b] David Derler, Sebastian Ramacher, and Daniel Slamanig. Short double- and n-times-authentication-preventing signatures from ECDSA and more. In *EuroS&P*, pages 273–287. IEEE, 2018.

[Fis99] Marc Fischlin. Pseudorandom function tribe ensembles based on one-way permutations: Improvements and applications. In *EUROCRYPT*, volume 1592 of *Lecture Notes in Computer Science*, pages 432–445. Springer, 1999.

[FKMV12] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the non-malleability of the fiat-shamir transform. In *INDOCRYPT*, volume 7668 of *Lecture Notes in Computer Science*, pages 60–79. Springer, 2012.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[GMO16] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. Zkboo: Faster zero-knowledge for boolean circuits. In *USENIX Security Symposium*, pages 1069–1083. USENIX Association, 2016.

[GQ88] Louis C. Guillou and Jean-Jacques Quisquater. A ”paradoxical” indentity-based signature scheme resulting from zero-knowledge. In *CRYPTO*, volume 403 of *Lecture Notes in Computer Science*, pages 216–231. Springer, 1988.

[IKOS09] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge proofs from secure multiparty computation. *SIAM J. Comput.*, 39(3):1121–1152, 2009.

[KKW18] Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In *ACM Conference on Computer and Communications Security*, pages 525–537. ACM, 2018.

[MR02] Silvio Micali and Leonid Reyzin. Improving the exact security of digital signature schemes. *J. Cryptology*, 15(1):1–18, 2002.

[Poe18] Bertram Poettering. Shorter double-authentication preventing signatures for small address spaces. In *AFRICACRYPT*, volume 10831 of *Lecture Notes in Computer Science*, pages 344–361. Springer, 2018.

[PS14] Bertram Poettering and Douglas Stebila. Double-authentication-preventing signatures. In *ESORICS (1)*, volume 8712 of *Lecture Notes in Computer Science*, pages 436–453. Springer, 2014.

[PS17] Bertram Poettering and Douglas Stebila. Double-authentication-preventing signatures. *Int. J. Inf. Sec.*, 16(1):1–22, 2017.

[RKS15] Tim Ruffing, Aniket Kate, and Dominique Schröder. Liar, liar, coins on fire!: Penalizing equivocation by loss of bitcoins. In *ACM Conference on Computer and Communications Security*, pages 219–230. ACM, 2015.

[Sch89] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 1989.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[Unr12] Dominique Unruh. Quantum proofs of knowledge. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 135–152. Springer, 2012.

[Unr15] Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *EUROCRYPT (2)*, volume 9057 of *Lecture Notes in Computer Science*, pages 755–784. Springer, 2015.

[Unr16] Dominique Unruh. Computationally binding quantum commitments. In *EUROCRYPT (2)*, volume 9666 of *Lecture Notes in Computer Science*, pages 497–527. Springer, 2016.

[Unr17] Dominique Unruh. Post-quantum security of fiat-shamir. In *ASIACRYPT (1)*, volume 10624 of *Lecture Notes in Computer Science*, pages 65–95. Springer, 2017.

A DSE* Security of DAPS

We recall the DSE^* security notion of DAPS. The game is depicted in Figure 6, where in contrast to Figure 3 the keys are generated by the adversary.

Definition 10 (DSE^* [PS14]). *For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of double-signature extraction under malicious keys (DSE^*) as*

$$\mathsf{Adv}^{\mathsf{DSE}^*}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = \Pr\left[\mathsf{Exp}^{\mathsf{DSE}^*}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = 1\right]$$

where the corresponding experiment is depicted in Figure 6. If for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{DSE}^*}_{\mathcal{A},\mathsf{DAPS}}(\kappa) \leq \varepsilon(\kappa),$$

then DAPS *provides* DSE^**.*

```
$\mathsf{Exp}^{\mathsf{DSE}^*}_{\mathcal{A},\mathsf{DAPS}}(\kappa)$:
    $(\mathsf{pk}_\mathsf{D}, m_1, m_2, \sigma_1, \sigma_2) \leftarrow \mathcal{A}(1^\kappa)$
    return 0, if $m_1$ and $m_2$ are not colliding
    return 0, if $\mathsf{Verify}_\mathsf{D}(\mathsf{pk}_\mathsf{D}, m_i, \sigma_i) = 0$ for any $i \in [2]$
    $\mathsf{sk}'_\mathsf{D} \leftarrow \mathsf{Ex}_\mathsf{D}(\mathsf{pk}_\mathsf{D}, m_1, m_2, \sigma_1, \sigma_2)$
    return 1, if $\mathsf{sk}'$ is not the secret key corresponding to $\mathsf{pk}_\mathsf{D}$
    return 0
```

Fig. 6: DSE^* **security for** DAPS.

B $\varSigma$-Protocols

Let $L \subseteq \mathsf{X}$ be an **NP**-language with associated witness relation R so that $L = \{x \mid \exists w : R(x, w) = 1\}$. A $\varSigma$-protocol for language L is defined as follows.

Definition 11. *A $\varSigma$-protocol for language L is an interactive three-move protocol between a PPT prover* $\mathsf{P} = (\mathsf{Commit}, \mathsf{Prove})$ *and a PPT verifier* $\mathsf{V} = (\mathsf{Challenge}, \mathsf{Verify})$*, where* P *makes the first move and transcripts are of the form* $(\mathsf{a}, \mathsf{c}, \mathsf{s}) \in \mathsf{A} \times \mathsf{C} \times \mathsf{S}$*. Additionally they satisfy the following properties:*

Completeness *A $\varSigma$-protocol for language L is complete, if for all security parameters κ, and for all $(x, w) \in R$, it holds that*

$$\Pr[\langle \mathsf{P}(1^\kappa, x, w), \mathsf{V}(1^\kappa, x)\rangle = 1] = 1.$$

Special Soundness *A $\varSigma$-protocol for language L is special sound, if there exists a PPT extractor $\mathcal{E}$ so that for all x, and for all sets of accepting transcripts $\{(\mathsf{a}, \mathsf{c}_i, \mathsf{s}_i)\}_{i\in[2]}$ with respect to x where $\mathsf{c}_1 \neq \mathsf{c}_2$, generated by any algorithm with polynomial runtime in κ, it holds that*

$$\Pr\left[w \leftarrow \mathcal{E}(1^\kappa, x, \{(\mathsf{a}, \mathsf{c}_i, \mathsf{s}_i)\}_{i\in[2]}) \ : \ (x, w) \in R\right] \geq 1 - \varepsilon(\kappa).$$

Special Honest-Verifier Zero-Knowledge *A $\varSigma$-protocol is special honest-verifier zero-knowledge, if there exists a PPT simulator $\mathcal{S}$ so that for every $x \in L$ and every challenge* c *from the challenge space, it holds that a transcript* $(\mathsf{a}, \mathsf{c}, \mathsf{s})$*, where* $(\mathsf{a}, \mathsf{s}) \leftarrow \mathcal{S}(1^\kappa, x, \mathsf{c})$ *is indistinguishable from a transcript resulting from an honest execution of the protocol.*

C NIZK Security Properties

Definition 12 (Completeness). *A non-interactive proof system for language L is complete, if for all $\kappa \in \mathbb{N}$, for all* $\mathsf{crs} \leftarrow \mathsf{Setup}_\Pi(1^\kappa)$*, for all $x \in L$, for all w such that $R(x, w) = 1$, and for all* $\pi \leftarrow \mathsf{Proof}_\Pi(\mathsf{crs}, x, w)$*, we have that* $\mathsf{Verify}_\Pi(\mathsf{crs}, x, \pi) = 1$*.*

This captures perfect completeness.

Definition 13 (Soundness). *For an efficient adversary $\mathcal{A}$, we define the advantage function in the sense of soundness as*

$$\mathsf{Adv}^{\mathsf{Sound}}_{\mathcal{A},\Pi}(\kappa) = \Pr\left[\begin{array}{l}\mathsf{crs} \leftarrow \mathsf{Setup}_\Pi(1^\kappa),\\ (x, \pi) \leftarrow \mathcal{A}(\mathsf{crs})\end{array} : \begin{array}{r}\mathsf{Verify}_\Pi(\mathsf{crs}, x, \pi) = 1\\ \wedge\ x \notin L\end{array}\right].$$

If for any efficient adversary $\mathcal{A}$ there exists a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{Sound}}_{\mathcal{A},\Pi}(\kappa) \leq \varepsilon(\kappa),$$

Π *is sound.*

Definition 14 (Adaptive Zero-Knowledge). *For an efficient simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ and an efficient adversary $\mathcal{A}$, we define the advantage functions in the sense of zero-knowledge as*

$$\mathsf{Adv}^{\mathsf{Sim}}_{\mathcal{A},\mathcal{S},\Pi}(\kappa) = \left|\begin{array}{l}\Pr\left[\mathsf{crs} \leftarrow \mathsf{Setup}_\Pi(1^\kappa) : \mathcal{A}(\mathsf{crs}) = 1\right] - \\ \Pr\left[(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^\kappa) : \mathcal{A}(\mathsf{crs}) = 1\right]\end{array}\right|$$

and

$$\mathsf{Adv}^{\mathsf{ZK}}_{\mathcal{A},\mathcal{S},\Pi}(\kappa) = \left|\Pr\left[\mathsf{Exp}^{\mathsf{ZK}}_{\mathcal{A},\mathcal{S},\Pi}(\kappa) = 1\right] - \frac{1}{2}\right|$$

where the corresponding experiment is depicted in Figure 7. If there exists an efficient simulator $\mathsf{S} = (\mathcal{S}_1, \mathcal{S}_2)$ *such that for any efficient adversary $\mathcal{A}$ there exist negligible functions $\varepsilon_1(\cdot)$ and $\varepsilon_2(\cdot)$ such that*

$$\mathsf{Adv}^{\mathsf{Sim}}_{\mathcal{A},\mathcal{S},\Pi}(\kappa) \leq \varepsilon_1(\kappa) \text{ and } \mathsf{Adv}^{\mathsf{ZK}}_{\mathcal{A},\mathcal{S},\Pi}(\kappa) \leq \varepsilon_2(\kappa)$$

then Π *provides adaptive zero-knowledge.*

Definition 15 (Simulation-Sound Extractability). *For an adaptively zero-knowledge non-interactive proof system* Π*, for an efficient extractor extractor $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$ and an efficient adversary $\mathcal{A}$, we define the advantage functions in the sense of simulation-sound extractability as*

$$\mathsf{Adv}^{\mathsf{Ext_1}}_{\mathcal{A},\mathcal{E},\Pi}(\kappa) = \left|\begin{array}{l}\Pr\left[(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^\kappa) : \mathcal{A}(\mathsf{crs}) = 1\right] - \\ \Pr\left[(\mathsf{crs}, \tau, \xi) \leftarrow \mathcal{E}_1(1^\kappa) : \mathcal{A}(\mathsf{crs}) = 1\right]\end{array}\right|$$

```
$\mathsf{Exp}^{\mathsf{ZK}}_{\mathcal{A},\mathsf{S},\Pi}(\kappa)$:
  $b \leftarrow \{0,1\}$
  $(\mathsf{crs},\tau) \leftarrow \mathcal{S}_1(1^\kappa)$
  $b^* \leftarrow \mathcal{A}^{\mathsf{P}_b(\cdot,\cdot)}(\mathsf{crs})$
    where oracle $\mathsf{P}_0$ on input $(x,w)$:
      return $\pi \leftarrow \mathsf{Proof}_\Pi(\mathsf{crs},x,w)$, if $(x,w) \in R$
      return $\perp$
    and oracle $\mathsf{P}_1$ on input $(x,w)$:
      return $\pi \leftarrow \mathcal{S}_2(\mathsf{crs},\tau,x)$, if $(x,w) \in R$
      return $\perp$
  return 1, if $b = b^*$
  return 0
```

Fig. 7: Adaptive Zero-Knowledge

and

$$\mathsf{Adv}^{\mathsf{Ext}_2}_{\mathcal{A},\mathcal{E},\Pi}(\kappa) = \Pr\left[\mathsf{Exp}^{\mathsf{Ext}_2}_{\mathcal{A},\mathcal{E},\Pi}(\kappa) = 1\right]$$

where the corresponding experiment is depicted in Figure 8. If there exists an efficient extractor $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$ such that for any efficient adversary $\mathcal{A}$ there exist negligible functions $\varepsilon_1(\cdot)$ and $\varepsilon_2(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{Ext}_1}_{\mathcal{A},\mathcal{E},\Pi}(\kappa) \leq \varepsilon_1(\kappa) \text{ and } \mathsf{Adv}^{\mathsf{Ext}_2}_{\mathcal{A},\mathcal{E},\Pi}(\kappa) \leq \varepsilon_2(\kappa)$$

then Π provides simulation-sound extractactability.

```
$\mathsf{Exp}^{\mathsf{Ext}_2}_{\mathcal{A},\mathcal{E},\Pi}(\kappa)$:
  $(\mathsf{crs},\tau,\xi) \leftarrow \mathcal{E}_1(1^\kappa)$
  $\mathcal{Q}_\mathcal{S} = \emptyset$
  $(x^*,w^*) \leftarrow \mathcal{A}^{\mathcal{S}(\cdot,\cdot)}(\mathsf{crs})$
    where oracle $\mathcal{S}$ on input $(x,w)$:
      $\mathcal{Q}_\mathcal{S} \leftarrow \mathcal{Q}_\mathcal{S} \cup \{(x,w)\}$
      return $\pi \leftarrow \mathcal{S}_2(\mathsf{crs},\tau,x)$, if $(x,w) \in R$
      return $\perp$
  $w \leftarrow \mathcal{E}_2(\mathsf{crs},\xi,x^*,\pi^*)$
  return 1, if $\mathsf{Verify}_\Pi(\mathsf{crs},x^*,\pi^*) = 1 \wedge (x^*,\pi^*) \notin \mathcal{Q}_\mathcal{S} \wedge (x^*,w) \notin R$
  return 0
```

Fig. 8: Simulation-sound extractability

C.1 NIZK from Σ-Protocols

To convert a Σ-protocol to a NIZK, $\mathsf{Setup}_\Pi(1^\kappa)$ fixes a hash function $H : \mathsf{A} \times \mathsf{X} \rightarrow \mathsf{C}$, sets $\mathsf{crs} \leftarrow (\kappa, H)$ and returns crs. The algorithms Proof_Π and Verify_Π are defined as follows:

$\mathsf{Proof}_\Pi(\mathsf{crs}, x, w)$: Start P on $(1^\kappa, x, w)$, obtain the first message a, answer with $\mathsf{c} \leftarrow H(\mathsf{a}, x)$. Finally obtain s and return $\pi \leftarrow (\mathsf{a}, \mathsf{s})$.

$\mathsf{Verify}_\Pi(\mathsf{crs}, x, \pi)$: Parse π as (a, s). Start V on $(1^\kappa, x)$ and send a as first message to the verifier. When V outputs c, reply with s and output 1 if V accepts and 0 otherwise.

Combining [FKMV12, Thm. 1, Thm. 2, Thm. 3, Prop. 1] (among others) shows that a so-obtained proof system is complete, sound, adaptively zero-knowledge, if the underlying Σ-protocol is special sound and the commitments sent in the first move are unconditionally binding. Security of the Fiat-Shamir transform in the quantum-accessible ROM (QROM) requires stronger properties of the Σ-protocols [Unr17], however Unruh's transform [Unr12,Unr15,Unr16] can be used to obtain QROM-secure NIZKs from Σ-protocols.

D One-way Functions and Pseudorandom Function Families

We recall the definitions of one-way functions and pseudorandom function (families).

Definition 16 (OWF). *Let $f : S \rightarrow P$ be a function. For a PPT adversary $\mathcal{A}$ we define the advantage function as*

$$\mathsf{Adv}^{\mathsf{OWF}}_{\mathcal{A},f}(\kappa) = \Pr\left[x \xleftarrow{R} S, x^* \leftarrow \mathcal{A}(1^\kappa, f(x)) : f(x) = f(\mathcal{A}^*)\right] .$$

The function f is one-way function (OWF) if it is efficiently computable and for all PPT adversaries $\mathcal{A}$ there exists a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{OWF}}_{\mathcal{A},f}(\kappa) \leq \varepsilon(\kappa).$$

Definition 17 (PRF). *Let $\mathcal{F} : \mathcal{S} \times D \rightarrow \mathsf{R}$ be a family of functions and let Γ be the set of all functions $D \rightarrow \mathsf{R}$. For a PPT distinguisher $\mathcal{D}$ we define the advantage function as*

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{D},\mathcal{F}}(\kappa) = \Big|\Pr\Big[s \xleftarrow{R} \mathcal{S}, \mathcal{D}^{\mathcal{F}(s,\cdot)}(1^\kappa) = 1] - \Pr[f \xleftarrow{R} \Gamma, \mathcal{D}^{f(\cdot)}(1^\kappa) = 1\Big]\Big| .$$

$\mathcal{F}$ is a pseudorandom function (family) if it is efficiently computable and for all PPT distinguishers $\mathcal{D}$ there exists a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{D},\mathcal{F}}(\kappa) \leq \varepsilon(\kappa).$$

Below, we provide a slightly stronger variant of a definition of a notion introduced in [CMR98,Fis99].

Definition 18 (Fixed-Value-Key-Binding PRF). *A PRF family $\mathcal{F} : \mathcal{S} \times D \rightarrow \mathsf{R}$ and a $\beta \in D$, is fixed-value-key-binding if for all adversaries $\mathcal{A}$*

$$\Pr\left[s \xleftarrow{R} \mathcal{S}, s' \leftarrow \mathcal{A}(s, \beta) : \mathcal{F}(s, \beta) = \mathcal{F}(s', \beta) \ \wedge \ s \neq s'\right] = 0.$$

Moreover, we present a relaxed (computational) version of the above definition.

Definition 19 (Computational Fixed-Value-Key-Binding PRF). *For a PRF family $\mathcal{F} : \mathcal{S} \times D \to \mathsf{R}$ and a $\beta \in D$, we define the advantage function of a PPT adversary $\mathcal{A}$ as*

$$\mathsf{Adv}^{\mathsf{cFKVB}}_{\mathcal{A},\mathcal{F}}(\kappa) = \Pr\left[s \xleftarrow{R} \mathcal{S}, s' \leftarrow \mathcal{A}(1^\kappa, s, \beta) : \mathcal{F}(s,\beta) = \mathcal{F}(s',\beta) \ \wedge \ s \neq s'\right].$$

$\mathcal{F}$ is computationally fixed-value-key-binding if for all PPT adversaries there exists as negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{cFKVB}}_{\mathcal{A},\mathcal{F}}(\kappa) = \varepsilon(\kappa).$$

E Security Proofs

E.1 Proof of Theorem 3

Proof. To prove the theorem above, we proceed in a sequence of games where we play $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{DAPS},\mathcal{A}}(\kappa)$ with the DAPS in Scheme 2 and adversary $\mathcal{A}$.

Game 0: The original unforgeability game.

Game 1: As Game 0, but we choose Y uniformly at random and abort as soon as $\mathcal{A}$ queries the random oracle H on sk with $\mathsf{VKey}(\mathsf{sk}, \mathsf{pk}) = 1$.

Transition $0 \Rightarrow 1$**:** Let this event be called E. The distributions in Game 0 and Game 1 are identical unless E happens. We bound the probability of E to happen by constructing an adversary $\mathcal{B}$ with

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B},\mathsf{DAPS}}(\kappa) \geq \Pr[E].$$

To do so, we honestly generate $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma)$ and engage in an experiment $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B},\mathsf{DAPS}}(\kappa)$ to obtain pk for DAPS. We choose Y uniformly at random, and set $(\mathsf{sk}_\mathsf{D}, \mathsf{pk}_\mathsf{D}) \leftarrow ((\mathsf{sk}_\Sigma, \bot), (\mathsf{pk}_\Sigma, \mathsf{pk}_\mathsf{D}, Y))$. Whenever a signature for DAPS is required, we use the signing oracle provided by $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B},\mathsf{DAPS}}(\kappa)$. If E happens, we have that $\mathsf{VKey}(\mathsf{sk}, \mathsf{pk}) = 1$, which—by the correctness of DAPS—means that we can choose an arbitrary unqueried message m from the message space of DAPS which satisfies the winning condition, and output $(m, \mathsf{DAPS.Sign}_\mathsf{D}(\mathsf{sk}, m))$ as a forgery for DAPS. All in all, we thus have that $|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B},\mathsf{DAPS}}(\kappa)$.

Reduction. Now we are ready to show that the winning probability in Game 1 is bounded by $\max\{\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_1,\Sigma}(\kappa), \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_2,\mathsf{DAPS}}(\kappa)\}$. To do so, we construct two reductions which use $\mathcal{A}$ to construct $\mathcal{B}_1$ or $\mathcal{B}_2$ respectively. Both $\mathcal{B}_1$ and $\mathcal{B}_2$ will succeed whenever $\mathcal{A}$ succeeds.

$\mathcal{B}_1$: In this case, we engage in an experiment $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_1,\Sigma}(\kappa)$ to obtain pk_Σ. We choose Y uniformly at random, obtain $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{DAPS.KGen}_\mathsf{D}(1^\kappa)$ and set $(\mathsf{sk}_\mathsf{D}, \mathsf{pk}_\mathsf{D}) \leftarrow ((\bot, \mathsf{sk}), (\mathsf{pk}_\Sigma, \mathsf{pk}, Y))$. Whenever a Σ signature is required, the signature is obtained using the oracle provided by the experiment. If the adversary eventually outputs a forgery $(m^*, \sigma^*) = (m^*, (\sigma_0^*, \sigma_1^*))$ we output (m^*, σ_0^*) as a forgery to win $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_1,\Sigma}(\kappa)$. Clearly, $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\text{Game 1}}(\kappa) \leq \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_1,\Sigma}(\kappa)$.

$\mathcal{B}_2$: In this case, we engage in an experiment $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_2,\mathsf{DAPS}}(\kappa)$ to obtain pk. We choose Y uniformly at random, obtain $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow \Sigma.\mathsf{KGen}_\Sigma(1^\kappa)$ and set $(\mathsf{sk}_\mathsf{D}, \mathsf{pk}_\mathsf{D}) \leftarrow ((\mathsf{sk}_\Sigma, \bot), (\mathsf{pk}_\Sigma, \mathsf{pk}, Y))$. Whenever a DAPS signature is required, the signature is obtained using the oracle provided by the experiment. If the adversary eventually outputs a forgery $(m^*, \sigma^*) = (m^*, (\sigma_0^*, \sigma_1^*))$ we output (m^*, σ_1^*) as a forgery to win $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_2,\mathsf{DAPS}}(\kappa)$. Clearly, $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\text{Game 1}}(\kappa) \leq \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_2,\mathsf{DAPS}}(\kappa)$.

All in all, we now have $\Pr[S_0] = \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) \leq \max\{\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_1,\Sigma}(\kappa), \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_2,\mathsf{DAPS}}(\kappa)\} + \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{DAPS},\mathcal{B}}(\kappa)$ which concludes the prove. □

E.2 Proof of Theorem 4

Proof. We prove this theorem using a reduction. Assume that $\mathcal{A}$ breaks DSE-security of Scheme 2. We build a DSE adversary $\mathcal{B}$ against DAPS: When $\mathcal{B}$ is started on the secret key sk and public key pk of DAPS, we compute the key pair of Σ honestly, i.e., $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow \Sigma.\mathsf{KGen}_\Sigma(1^\kappa)$. Then, we compute the combined public key by extending it with $Y \leftarrow \mathsf{sk}_\Sigma \oplus H(\mathsf{sk})$. Now, we start $\mathcal{A}$ on the combined key-pair $(\mathsf{sk}_\Sigma, \mathsf{sk}), (\mathsf{pk}_\Sigma, \mathsf{pk}, Y)$. Once $\mathcal{A}$ returns colliding messages m_1, m_2 and signatures $\sigma_1 = (\sigma_{1,0}, \sigma_{1,1})$, $\sigma_2 = (\sigma_{2,0}, \sigma_{2,1})$, forward the messages with the corresponding DAPS signatures $\sigma_{1,1}, \sigma_{2,1}$ to $\mathcal{B}$. Let $(\mathsf{sk}^*_\Sigma, \mathsf{sk}^*) \leftarrow \mathsf{Ex}_\mathsf{D}((\mathsf{pk}_\Sigma, \mathsf{pk}, Y), m_1, m_2, \sigma_1, \sigma_2)$. Since, by definition, the adversary needs to output $(\mathsf{sk}^*_\Sigma, \mathsf{sk}^*) \neq (\mathsf{sk}_\Sigma, \mathsf{sk})$, it follows that $\mathsf{sk}^*_\Sigma \neq \mathsf{sk}_\Sigma$ or $\mathsf{sk}^* \neq \mathsf{sk}$. If we have $\mathsf{sk}^* = \mathsf{sk}$, we have that $\mathsf{sk}^*_\Sigma = Y \oplus H(\mathsf{sk}) = \mathsf{sk}_\Sigma$ since Y was set up honestly. Hence we have $\mathsf{sk}^* \neq \mathsf{sk}$, so $\mathcal{B}$ wins the DSE-security game if $\mathcal{A}$ wins it, which concludes the proof. □