Signatures on Equivalence Classes: A New Tool for Privacy-Enhancing Cryptography

by

Christian Hanser

A PhD Thesis
Presented to the Faculty of Computer Science and Biomedical Engineering in Partial Fulfillment of the Requirements for the PhD Degree

Assessors

Prof. Dr. Christian Rechberger (Graz University of Technology, Austria)
Prof. Dr. Sebastian Faust (Ruhr University Bochum, Germany)

February 2016

Institute for Applied Information Processing and Communications (IAIK)
Faculty of Computer Science and Biomedical Engineering
Graz University of Technology, Austria

Abstract

During the past decades the world has seen a rapid and historically unprecedented evolvement of technology. Computer and Internet technology, in particular, have penetrated practically all spheres of life. Meanwhile privacy has become a rare good, since privacy and data security concerns have often been put aside. The reasons are to seek in cost efficiency or the business model, in plain indifference or lack of awareness of users and sometimes in the complexity and practical inefficiency of cryptographic solutions. We deliberately outsource large amounts of personal data to places we do not know and cannot control through communication channels of sometimes questionable security. At the same time, our actions leave digital footprints, whose extents we can barely conceive. A fortiori, it is a necessity to counter these developments through strong privacy-enhancing cryptography, which allows us to secure our personal data and reduce our communication traces that are analyzable by third parties while still upholding functionality.

This thesis introduces structure-preserving signatures on equivalence classes (SPS-EQs) and presents several applications to privacy-enhancing cryptography. Loosely speaking, an SPS-EQ allows us to sign projective equivalence classes (defined in the bilinear-group setting) and to adapt signatures to arbitrary representatives of the respective class later on. At the same time, it should be infeasible to link message-signature pairs belonging to the same class. Surprisingly, SPS-EQs enable new construction paradigms for very efficient and intelligible schemes. We will describe two SPS-EQ constructions and a security model, which we will also discuss in more detail. Using SPS-EQs, we will then show a new and efficient way to build blind and partially blind signatures—two basic building blocks for privacy-enhancing protocols—, introduce new design paradigms for one-show attribute-based credentials (ABCs) and multi-show ABCs—methods that allow us to authenticate ourselves without disclosing our identity. More precisely, we will give the first practically efficient round-optimal blind-signature scheme having security proofs in the standard model. We will then show the first one-show ABC based on a standard-model blind-signature scheme. Further, we will present an efficient multi-show ABC along with a game-based security model and a new perfectly hiding set-commitment scheme as its second building block—both latter contributions are of independent interest. Our multi-show ABC is the first to simultaneously have constant-size credentials and constant communication effort—two distinguishing features for efficiency and thus practicality. Furthermore, it is the first ABC whose anonymity holds against malicious organization keys in the standard model. Last but not least, we will also take

a look at verifiably-encrypted signatures (VESs). These are signature schemes for fair exchange in digital business processes. We will point out flaws in their security model, show how to fix them and give a black-box VES construction from SPS-EQ, which allows us to relate SPS-EQs to public-key encryption and separate certain classes of SPS-EQ from one-way functions (OWFs). This relation is somewhat surprising, since digital signature schemes can usually be built from OWFs.

Kurzfassung

Während der letzten Jahrzehnte hat sich unsere Technologie sehr rapide und in historisch noch nie dagewesener Weise entwickelt. Computer- und speziell Internettechnologie sind in nahezu alle Sphären des täglichen Lebens eingedrungen. Da Datenschutz und Datensicherheit in den letzten Jahren sehr oft vernachlässigt wurden, ist unsere Privatsphäre mehr und mehr zu einem seltenen Gut avanciert. Die Gründe dafür sind vielschichtig und finden sich oft in der unaufhörlichen Suche nach Kosteneffizienzsteigerung, in den Geschäftsmodellen großer Konzerne, in purer Indifferenz oder einem Mangel an Bewusstsein der User, aber auch in der Komplexität und Impraktikabilität kryptografischer Lösungen. Wir alle lagern leichtfertig große Teile unserer perunseressönlichen Daten an Orte aus, die wir weder kennen noch kontrollieren können, und all das über Kommunikationskanäle von oft fragwürdiger Sicherheit. Gleichzeitig hinterlässt unser Verhalten digitale Fußabdrücke, deren Ausmaße schwer zu fassen sind. Damit wächst auch die Notwendigkeit, diesen Entwicklungen mit starken datenschutzfördernden Technologien (*privacy-enhancing technologies*) entgegenzutreten, welche es uns einerseits erlauben, unsere persönlichen Daten abzusichern und andererseits auch helfen, die digitalen Spuren, die wir alle tagtäglich hinterlassen und welche aktiv von Dritten analysiert werden, effektiv zu reduzieren und das bei gleichzeitiger Aufrechterhaltung des gewohnten Komforts.

Diese Arbeit führt neue digitale Signaturen namens *structure-preserving signatures on equivalence classes (SPS-EQs)* ein und präsentiert gleich mehrere Anwendungen von SPS-EQ zu *privacy-enhancing cryptography.* Ein SPS-EQ erlaubt uns, grob gesprochen, das Signieren projektiver Äquivalenzklassen (welche wir im Kontext von bilinearen Gruppen definieren) und in weiterer Folge das Ableiten von Signaturen zu beliebigen Repräsentanten entsprechender Klassen. Gleichzeitig soll es nicht möglich sein zwei Nachrichten-Signatur-Paare, welche in die selbe Klasse fallen, miteinander in Verbindung zu bringen. Interessanterweise ermöglichen diese Signaturen neue Konstruktionsweisen für sehr effiziente und gut verständliche kryptografische Schemen. Wir werden zwei SPS-EQ Konstruktionen und ein Sicherheitsmodell näher beschreiben und diskutieren. Weiters verwenden wir SPS-EQ, um blinde und partiell blinde Signaturen auf neue und sehr effiziente Art und Weise zu bauen. Dabei handelt es sich um grundlegende Bausteine für viele datenschutzfördernde Protokolle. Danach werden wir (aufbauend auf SPS-EQ) neue Methoden zur Konstruktion von *one-show* und *multi-show attribute-based credentials (ABCs)* aufzeigen – beides Methoden zur anonymen Authentifizierung. Im Besonderen geben wir das erste praktikable rundenoptimale blinde Signaturverfahren, welches Beweise

im Standardmodell besitzt. Darauf aufbauend führen wir das erste *one-show ABC* ein, das auf einem blinden Signaturverfahren im Standardmodell basiert. Weiters präsentieren wir ein sehr effizientes *multi-show ABC* zusammen mit einem spielbasierten Sicherheitsmodell als auch ein neues perfekt verbergendes Commitmentschema als zweiten Grundbaustein. Es sei angemerkt, dass die beiden letzteren Beiträge auch unabhängig von der *multi-show ABC* Konstruktion interessant sind. Unser *multi-show ABC* ist das erste solche Verfahren, welches gleichzeitig konstanten Kommunikationsaufwand und Credentials von konstanter Größe besitzt. Außerdem ist es das erste ABC, dessen Anonymitätsgarantien gegenüber böswillig erzeugten Organisationsschlüsseln im Standardmodell halten. Zu guter Letzt widmen wir uns dem Thema *verifiably-encrypted signatures (VESs)*. Diese Signaturverfahren ermöglichen den fairen Austausch in digitalen Geschäftsprozessen. Zum einen diskutieren wir mehrere Probleme in den etablierten Sicherheitsmodellen und zum anderen präsentieren wir eine generische (*black-box*) VES-Konstruktion basierend auf SPS-EQ. Diese erlaubt es uns in weiterer Folge, eine Verbindung zwischen SPS-EQ und *public-key encryption (PKE)* herzustellen, was es uns wiederum ermöglicht, bestimmte SPS-EQ-Klassen von Einwegfunktionen zu separieren. Dieser Zusammenhang ist ein wenig überraschend, da digitale Signaturen üblicherweise über Einwegfunktionen konstruiert werden können.

Acknowledgements

First of all, I want to thank my family, my friends and all the people who supported and challenged me during the last couple of years. Special thanks go to my colleague and friend Daniel Slamanig, who introduced me to provable security, shared his comprehensive knowledge and gave me the necessary directions to start and to accomplish my PhD. A big thanks goes out to my co-author Georg Fuchsbauer, especially for the patience and for the many things I learned from him. Further, I would like to thank my current supervisor Christian Rechberger for the valuable comments and both Christian and my former supervisor Roderick Bloem for the support and the counseling. I would like to thank my assessor Sebastian Faust for taking the time to read this thesis and give important feedback and, last but not least, for the readiness to travel to Graz. Special thanks go to Dominique Schröder, for the fruitful discussions and for inviting me to a research visit to Saarbrücken during which one publication originated. Finally, I want to thank all my co-authors and all my colleagues I worked with during this time: For the good time, for making work exciting and fun.

Christian Hanser
Graz, February 2016

Table of Contents

List of Publications

In Refereed Conference Proceedings

1. David Derler, Christian Hanser, Henrich C. Pöhls and Daniel Slamanig. *Towards Authenticity and Privacy Preserving Accountable Workflows.* In *Privacy and Identity Management for the Future Internet in the Age of Globalisation*, in press.

2. David Derler, Christian Hanser and Daniel Slamanig. *A New Approach To Efficient Revocable Attribute-Based Anonymous Credentials.* In *Cryptography and Coding - IMACC 2015 - 15th IMA International Conference on Cryptography and Coding, Oxford, UK, December 15–17, 2015. Proceedings*, volume 9496 of *Lecture Notes in Computer Science*, pages 57–74. Springer, 2015.

3. Christian Hanser, Max Rabkin and Dominique Schröder. *Verifiably Encrypted Signatures: Security Revisited and a New Construction.* In *Computer Security - ESORICS 2015 - 20th European Symposium on Research in Computer Security, Vienna, Austria, September 21–25, 2015. Proceedings*, volume 9326 of *Lecture Notes in Computer Science*, pages 146–164. Springer, 2015.

4. Georg Fuchsbauer, Christian Hanser and Daniel Slamanig. *Practical Round-Optimal Blind Signatures in the Standard Model.* In *Advances in Cryptology - CRYPTO (2) 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16–20, 2015. Proceedings*, volume 9216 of *Lecture Notes in Computer Science*, pages 233–253. Springer, 2015.

5. David Derler, Christian Hanser and Daniel Slamanig. *Revisiting Cryptographic Accumulators, Additional Properties and Relations to other Primitives.* In *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.

6. David Derler, Christian Hanser and Daniel Slamanig. *Blank Digital Signatures: Optimization and Practical Experiences.* In *Privacy and Identity Management for the Future Internet in the Age of Globalisation*, volume 457 of *IFIP Advances in Information and Communication Technology*, pages 201–215. Springer, 2015.

7. Christian Hanser and Daniel Slamanig. *Structure-Preserving Signatures on Equivalence Classes and their Application to Anonymous Credentials.* In *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan (R.O.C.), December 7–11, 2014. Proceedings*, volume 8873 of *Lecture Notes in Computer Science*, pages 491–511. Springer, 2014.

8. David Derler, Christian Hanser and Daniel Slamanig. *Privacy-Enhancing Proxy Signatures from Non-Interactive Anonymous Credentials.* In *28th Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy (DBSec 2014), Vienna, Austria, July 14–16, 2014. Proceedings*, volume 8566 of *Lecture Notes in Computer Science*, pages 49–65. Springer, 2014.

9. Christian Hanser and Daniel Slamanig. *Warrant-Hiding Delegation-by-Certificate Proxy Signature Schemes.* In *Progress in Cryptology - INDOCRYPT 2013, 14th International Conference on Cryptology in India, Mumbai, India, December 7–10, 2013. Proceedings*, volume 8250 of *Lecture Notes in Computer Science*, pages 60–77. Springer, 2013.

10. Christian Hanser and Christian Wagner. *Speeding Up the Fixed-Base Comb Method for Faster Scalar Multiplication on Koblitz Curves.* In *Security Engineering and Intelligence Informatics - CD-ARES 2013 Workshops: MoCrySEn and SeCIHD, Regensburg, Germany, September 2–6, 2013. Proceedings*, volume 8128 of *Lecture Notes in Computer Science*, pages 168–179. Springer, 2013.

11. Gerwin Gsenger and Christian Hanser. *Improving the Efficiency of Elliptic Curve Scalar Multiplication using Binary Huff Curves.* In *Security Engineering and Intelligence Informatics - CD-ARES 2013 Workshops: MoCrySEn and SeCIHD, Regensburg, Germany, September 2–6, 2013. Proceedings*, volume 8128 of *Lecture Notes in Computer Science*, pages 155–167. Springer, 2013.

12. Christian Hanser and Daniel Slamanig. *Efficient Simultaneous Privately and Publicly Verifiable Robust Provable Data Possession from Elliptic Curves.* In *10th International Conference on Security and Cryptography (SECRYPT 2013), Reykjavik, Iceland, 29–31 July 2013. Proceedings*, pages 15–26. SciTePress, 2013.

13. Klaus Potzmader, Johannes Winter, Daniel Hein, Christian Hanser, Peter Teufl and Liqun Chen. *Group Signatures on Mobile Devices: Practical Experiences.* In *Trust and Trustworthy Computing - 6th International Conference, TRUST 2013, London, UK, June 17–19, 2013. Proceedings*, volume 7904 of *Lecture Notes in Computer Science*, pages 47–64. Springer, 2013.

14. Christian Hanser and Daniel Slamanig. *Blank Digital Signatures*. In *8th ACM Symposium on Information, Computer and Communications Security, ASIACCS'13, Hangzhou, China, May 08–10, 2013. Proceedings*, pages 95–106. ACM, 2013.

15. Daniel Slamanig and Christian Hanser. *On Cloud Storage and the Cloud of Clouds Approach*. In *7th International Conference for Internet Technology and Secured Transactions, ICITST 2012, London, United Kingdom, December 10–12, 2012. Proceedings*, pages 649–655. IEEE, 2012.

In Review

1. Georg Fuchsbauer, Christian Hanser and Daniel Slamanig. *Structure-Preserving Signatures on Equivalence Classes and their Application to Anonymous Credentials*. Submitted to Journal of Cryptology, 2016.

Preprints

1. Georg Fuchsbauer, Christian Hanser and Daniel Slamanig. *EUF-CMA-Secure Structure-Preserving Signatures on Equivalence Classes*. Cryptology ePrint Archive, 2014/944, 2014. `http://eprint.iacr.org/`

2. Christian Hanser. *Performance of the SHA-3 Candidates in Java*. In *The Third SHA-3 Candidate Conference, Washington D.C., March 22–23, 2012.*

Master's Thesis

1. Christian Hanser, *New Trends in Elliptic Curve Cryptography*. Master's thesis, Institute for Applied Information Processing and Communications (IAIK), Graz University of Technology, Inffeldgasse 16a, 8010 Graz, Austria, May 2010.

List of Tables

List of Schemes

Acronyms

ABC attribute-based credential. iii, 3, 6–8, 10, 12, 14–18, 42, 55, 95, 101, 102, 104–106, 108–111, 114, 121, 125–127

BS blind signature. 61–64

CDH computational Diffie-Hellman. 10, 26, 27, 56, 89, 91

CRS common-reference string. 10–12, 14, 16, 23, 24, 37, 61, 105, 111, 121, 123

DDH decisional Diffie-Hellman. 7, 13, 14, 27, 42, 45, 46, 55–59, 68, 73, 74, 89, 91, 104, 119, 120

DHI Diffie-Hellman inversion. 26, 65–68, 74–76, 88, 89, 91

DL discrete logarithm. 11, 12, 25, 27, 30, 31, 33, 35, 36, 42, 65, 74, 95, 97, 109, 110, 112, 115, 116

DLIN decision linear. 26, 42, 56, 59

DLP discrete-logarithm problem. 56, 114, 117

DS digital signature. 13, 20, 38, 39, 126

ECC elliptic-curve cryptography. 4, 17

ECDLP elliptic-curve discrete-logarithm problem. 4

EUF-CMA existentially unforgeable under adaptive chosen-message attacks. 38–40, 43, 46, 47, 49–51, 55–59, 66, 67, 74–76, 84, 87, 88, 90, 92, 112, 114, 115, 129

GGM generic-group model. 8, 14, 23, 24, 28, 40, 41, 47, 49, 51, 55, 74, 125

GS Groth-Sahai. 5, 8, 10–12, 39, 46

HVZK honest-verifier zero-knowledge. 35

IND-CPA indistinguishability against chosen-plaintext attacks. 29, 31, 92

IPS interactive proof system. 31, 32, 34

NIZKP non-interactive zero-knowledge proof. 10, 11, 35

OWF one-way function. iv, 13, 17, 20, 21, 37, 80, 121, 126

PBS partially blind signature. 64, 65, 75

PKC public-key cryptography. 4

PKE public-key encryption. iv, vi, 13, 17, 19, 20, 29, 80, 82, 90–92

PKI public-key infrastructure. 38

PoK proof of knowledge. 12, 32–34, 75, 108, 109, 121

PPE pairing-product equation. 5, 8, 39, 40, 43, 47, 50, 55, 86

PPT probabilistic-polynomial time. 20–23, 25–27, 29–34, 38, 39, 42–44, 50, 56, 59, 62–64, 68, 80–82, 90, 93–95, 105, 108

RO random oracle. 8, 14, 61

ROM random-oracle model. 10, 11, 13, 15, 23, 24, 35, 39, 102

SDH strong Diffie-Hellman. 26, 28, 95, 97, 98

sEUF-CMA strongly existentially unforgeable under adaptive chosen-message attacks. 39

SPS structure-preserving signature. 5, 7, 8, 10, 11, 14, 39–41, 46, 47, 55, 57, 129

SPS-EQ structure-preserving signature on equivalence classes. iii–vi, ix, xvii, 7, 13–18, 41–51, 55–57, 59, 61, 65–69, 74–76, 79, 80, 85–92, 102, 104, 108–111, 114, 122, 125–127

SXDH Symmetric eXternal Diffie-Hellman. 27, 122, 123

TTP trusted third party. 24, 61

VES verifiably-encrypted signature. iv, vi, 4, 6–8, 13, 14, 17, 42, 55, 79–89, 91, 92, 125, 126

WI witness indistinguishability. 33, 36, 37

WIPoK witness-indistinguishable proof of knowledge. 33, 34

XDH eXternal Diffie-Hellman. 27, 122, 123

ZK zero knowledge. 17, 32–34, 37, 102, 103, 105, 115, 116, 118

ZKP zero-knowledge proof. 10, 15, 16, 32, 103

ZKPoK zero-knowledge proof of knowledge. 12, 15, 16, 34, 36, 37, 103–105, 109, 110, 112, 114–118, 120–122

Notation

$*$	A placeholder.
$\exists$	The existential quantifier.
$\forall$	The all quantifier.
$\langle \cdot, \cdot \rangle$	The inner product.
$\langle (\mathcal{P}, \mathcal{V}) \rangle$	The transcript of the interaction of two algorithms $\mathcal{P}$ and $\mathcal{V}$.
$[C]$	The Iverson bracket (returns 1 if condition C is true and 0 else).
$\langle P \rangle$	The group generated by some element P.
$\oplus$	The exclusive-or operator.
$\lvert \cdot \rvert$	Either the absolute value, the length of a bitstring or the cardinality of a set (or, more specifically, the order of a group).
$\top$	Indicates a successful termination.
$\bot$	Indicates an unsuccessful termination or an undefined value.
$\emptyset$	The empty set.
$\approx$	Identically distributed.
$\{0,1\}^n$	The set of all binary strings of length n.
$\{0,1\}^*$	The set of all binary strings of arbitrary length.
$0_{\mathbb{G}}$	The neutral element of an additive group $(\mathbb{G}, +)$.
$1_{\mathbb{G}}$	The neutral element of a multiplicative group $(\mathbb{G}, \cdot)$.
$a \leftarrow b$	Indicates an assignment of b to a.
$a \xleftarrow{R} A$	Drawing a value a uniformly at random from a set A.
$a \in_R A$	Indicates that a is uniformly random in set A.
$[n]$	The interval $[1, n]$ of natural numbers.
BG	A (Type-3) bilinear group.
BG_i	A Type-i bilinear group for $i \in [3]$.
$e(\cdot, \cdot)$	A bilinear map or pairing.
$\epsilon(\cdot)$	A negligible function.
$\mathbf{Exp}$	A random experiment.
$\mathbb{G}$	A group.
$\mathbb{G}^*$	The set of elements of group $\mathbb{G}$ without its neutral element $0_{\mathbb{G}}$.
$\mathbf{g}$	The generator of group $\mathbb{G}_T$.
κ	A security parameter.
$L[i]$	The ith element of a list L.
$\mathbb{N}$	The natural numbers.
$\mathcal{O}$	An oracle.
P	The generator of a group (typically of group $\mathbb{G}$ or $\mathbb{G}_1$).
$\hat{P}$	The generator of group $\mathbb{G}_2$.
pk	A public key.

pp	A set of public parameters.
$\Pr$	The probability measure.
$\mathcal{R}$	A (binary) relation.
$\mathbb{R}$	The real numbers.
$\mathbb{R}^+$	The positive real numbers.
sk	A private key.
$\vec{x}$	A vector.
$\mathbb{Z}$	The ring of integers.
$\mathbb{Z}_p$	The field of integers modulo a prime p.
$\mathbb{Z}_p^*$	The multiplicative subgroup of $\mathbb{Z}_p$.

1 Introduction

We were able to do a whole bunch of other things. Some of the other things were metadata, and bulk collection and so on.
— former NSA Director Michael Hayden, on data collection in front of encrypted communication

We kill people based on metadata.
— Michael Hayden

I believe there is something out there watching us. Unfortunately, it's the government.
— Woody Allen

Over the last couple of decades information technology has been evolving at a tremendous pace and been invading nearly all spheres of life. At the same time, data security and privacy have been neglected or often not been taken into account at all—may it be for deliberate reasons or for a lack of better knowledge. This has largely enabled a society where companies collect data about their customers, governments accumulate data about their citizens and individuals spy on each other—and all of this is happening on a historically unprecedented, vast scale. The dream of every 20th century intelligence agency has come true, privacy-undermining technologies such as data mining and big data have become a reality and the trade with personal data a lucrative business model—while (the human right for) seeking privacy is being interpreted more and more as suspicious behavior or even as a punishable act. Also, cloud computing, which offers

seemingly unlimited scalability and resources, has become a multi-billion dollar business model—mostly without taking account of data security and privacy. Interestingly, not only do individuals carelessly outsource increasingly large parts of their data, but so do also many companies without taking care of proper data protection. Ultimately, this affects the privacy of clients and employees and has potentially business-threatening consequences, such as industrial espionage and embarrassing data breaches. With the accumulation of data in the cloud, data breaches have become total and more devastating than ever before. Nevertheless, such events are not limited to the cloud alone: History has proven again and again that unprotected data is never safe. At this point, it must be made clear that danger is not only posed by content data, but also by collected metadata. This applies even to metadata gathered from encrypted communication. When linkable to individuals, metadata are a powerful means to draw comprehensive conclusions about them. Privacy-enhancing technologies offer protection for both types of data, that is, to keep our data safe and confidential and to maintain control over it. Indeed, a multitude of recent revelations have shown that privacy in this dynamic and increasingly interconnected new world cannot be enforced without being backed by strong mathematical security guarantees. The key to tackle this issue are privacy-enhancing technologies and, in particular, their strongest form: Privacy-enhancing cryptography. Privacy-enhancing cryptography gives us control over personal data and minimizes the amount of metadata and personal data that is collectable and analyzable by third parties. It aims at enforcing anonymity and unlinkability in digital interactions while it tries to maintain the comfort to which we got used in our digital age.

Many key technologies (some of them have already been envisioned by David Chaum [Cha81, Cha82, Cha85, Cha86] and Michael O. Rabin [Rab05] as of the early 1980s) are already available—at least in theory. Privacy-enhancing technologies that are in general use are, for instance, data encryption, which provides data confidentiality; and transport encryption (e.g., TLS [DR08]), which allows us to establish authentic and confidential communication channels. Other concepts in wider use are off-the-record (OTR) messaging [RGK05] (e.g., via Signal/Textsecure), which provides us with means for secure and confidential instant messaging; onion routing [RSG98] (e.g., Tor) and mix networks [Cha81] (e.g., Mixmaster), which help us in anonymizing our network traffic for low and high latency requirements, respectively; and modern e-cash systems [Cha82] (such as Bitcoin [Nak09]), which allow for (up to some extent) anonymous financial transactions.[1] A technology that is expected to come into play within the next couple of years is anonymous authentication, realized via anonymous credentials. They give us the possibility to authenticate ourselves anonymously in front of different organizations [Cha86, Bra00, LRSW99, CL01, BP10]—usually by presenting attributes about ourselves (e.g., our age), but not giving away any information beyond. Other privacy-enhancing solutions, which

[1] With regard to Bitcoin, various extensions and alternative approaches (such as Zerocoin [MGGR13], Zerocash [BCG+14], Spacecoin [DFKP15] or Spacemint [PPK+15]) target at increasing the efficiency or the level of anonymity.

do not yet see practical deployment, are, for example, private information retrieval (PIR) [KO97] and oblivious RAM [GO96], which hide data access patterns; searchable encryption [CGKO06], which enables searches on encrypted data without allowing the search engine to learn neither the keywords nor the plaintext; and (fully) homomorphic encryption [RAD78, Gen09], which enables third parties to perform (arbitrary) computations on encrypted data without learning anything about the plaintext and, in doing so, allows the outsourcing of computations on confidential data into the cloud.

So far, many more sophisticated privacy-enhancing technologies have not really found their ways into our daily lives. At the same time, it would have never been so easy to deploy secure and privacy-preserving technologies—given the widespread dissemination and omnipresence of smartphones and gadgets. The main reasons blocking many privacy-enhancing technologies from prime-time use are (1) in some cases practical inefficiency, (2) the often high complexity and bad accessibility for implementers, (3) sometimes simply the poor usability of respective applications and (4) the widespread unawareness about negative consequences of mass surveillance (and the lack of knowledge how to oppose it) on the one hand and poor negligence (or even digital promiscuity) on the other hand.

Facing these threats from the viewpoint of cryptography, it is crucial to address issues (1) and (2) and bring security and privacy technologies closer to practice—particularly through the design of practically efficient, intelligible and usable technologies. A major part of this thesis is dedicated to this purpose.[2] It gives, in fact, new construction paradigms for privacy-enhancing schemes that help us reducing collectable metadata while upholding functionality. In particular, it aims at providing intelligible and at the same time practically (and indeed highly) efficient new approaches to anonymous authentication and other fundamental privacy-protecting techniques. We will not only propose several new approaches to build anonymous-authentication protocols (in the form of one-show and multi-show attribute-based credentials (ABCs)), blind signatures (which form a basis for one-show attribute-based credentials (ABCs), e-voting and e-cash), but we will also develop and study the required basic building blocks. To this end, we will introduce some new and surprisingly versatile cryptographic primitives, which are also of independent interest. Still, it has to be noted that more work on these building blocks is still necessary in order to achieve even stronger security guarantees.

1.1 Background

We will start this section with a general discussion of the cryptographic setting, in which we will operate, followed by an outline of the basic ideas behind the cryptographic schemes and protocols we are going to consider in this thesis.

[2] The remaining parts deal with security in electronic-business processes and, most importantly, their results allow further insights on the basic building blocks that we are going to introduce in the course of this thesis and on which we will base the other central results.

1.1.1 Pairing-Based Cryptography

The techniques used throughout this thesis are based on bilinear groups, which can be efficiently set up in the elliptic curve setting. Elliptic curves have been studied in mathematics for a long time. The set of points on an elliptic curve (together with the point at infinity) forms a group, where the group law is given by the so-called *chord-and-tangent method* (for further details we refer the reader to, e.g., [Sil86, HMV03, CF05]). Their applications are manyfold and range from the study of integrals to the proof of Fermat's last theorem by Andrew Wiles [Wil95]. In the 1980s, they were introduced to cryptography by Miller and Koblitz [Mil86b, Kob89]. Since then elliptic curves have become an important pillar of modern public-key cryptography (PKC) and an efficient alternative to other cryptosystems like RSA [RSA78]. The great benefit of elliptic curves lies in the fact that no sub-exponential-time algorithm for solving the elliptic-curve discrete-logarithm problem (ECDLP) is known. Compared to RSA, for which algorithms solving the underlying integer-factorization problem (IFP) in sub-exponential time are known (e.g., the number field sieve [Pol93]), this facilitates the use of by far smaller key sizes and, thus, more efficient implementations—despite the more complex arithmetic. Pairings on elliptic curves (or more generally on Abelian varieties) were first defined by Weil [Wei40] as early as in the year 1940 and have for a quite a while been conceived as a purely theoretical concept. In the context of elliptic curves, a pairing is a bilinear map mapping two group elements to a target group.[3] In the 1990s their first application to elliptic-curve cryptography (ECC) was found, namely, a negative one: An attack against the ECDLP on certain curves equipped with an efficiently computable bilinear map. Even though, this event shattered the trust in ECC for a short period in the 1990s, it did, interestingly enough, neither block the rise of ECC in the long run nor the advent of pairing-based cryptography. However, from then on, it would take until the early 2000s to discover positive applications of bilinear maps, such as one-round tripartite Diffie-Hellman key agreement [Jou00] or identity-based encryption (IBE) [Sha84, BF01], and become aware of their huge potential. Since then pairing-based cryptography and with it cryptography itself has undergone considerable development. Several breakthroughs enabled the efficient computation of pairings (e.g., [Mil86a, MNT01, BKLS02, BN06, Ver10]) and further triggered new directions and rapidly evolving new developments in cryptography [Jou00, BF01, BLS01]. These advances led to a blossoming of public-key cryptography that has been going on until today. Besides enabling solutions to long-standing open problems (such as IBE [Sha84, BF01]), it gave rise to a plethora of new cryptographic primitives (such as short aggregate signatures [BLS01], verifiably-encrypted signatures (VESs) [BGLS03], attribute-based encryption [GPSW06] and the like) and enabled more efficient and more elegant

[3] There are several types of pairings: Type-1, Type-2 and Type-3 pairings. In the Type-1 (or symmetric) setting, the pairing maps two elements coming from the same group to a target group. In the Type-2 and Type-3 (or asymmetric) settings, the pairing maps two elements stemming from two related, yet distinct, groups to a target group. Moreover, in the Type-3 setting those two base groups are strictly separated. For more details, we refer the reader to Section 2.3.

constructions of already known cryptographic primitives (e.g., short group signatures [BBS04] and blind signatures [Bol03]). Due to such theoretical breakthroughs and the hunt for practically efficient pairing implementations, which reached its climax in the development of the optimal Ate pairing [Ver10] on Barreto-Naehrig (BN) curves [BN06], pairing-based cryptography has become one of the most important building blocks of modern cryptography.[4]

Another significant breakthrough was the introduction of the Groth-Sahai (GS) proof system in 2007 [GS07, GS08], which is an efficient framework for non-interactive witness-indistinguishable (NIWI) and non-interactive zero-knowledge (NIZK) proofs for languages expressible by bilinear groups. This led to the development of structure-preserving signatures (SPSs) [AFG+10]—a new signature scheme type compatible with GS proofs. In particular, SPSs are defined on bilinear groups, that is, messages, public keys and signatures are group elements and verification is performed by a conjunction of pairing-product equations (PPEs) and group membership tests.

Current cutting-edge research strives to develop multilinear maps [BS02a], in order to continue and generalize the ideas of pairing-based cryptography. Right now, many new developments follow in quick succession: Most of all, we see big advances on indistinguishability obfuscation (iO) (e.g., [GGH+13b, GH13a, PS15a, AFH+15]), which is equivalent to multilinear maps [PS15a, AFH+15], and the quest for multilinear maps constructions (cf., e.g., [GGH13a, CLT13, LSS14, CLT15, GGH15]) alternating with the chase to break them (cf., e.g., [GHMS14, CLT14, CLR15, CLLT15, CFL+16]).

1.1.2 Blind Signatures

The concept of blind signatures [Cha82] dates back to the beginning of the 1980s. A blind-signature scheme is an interactive protocol where a user (or obtainer) requests a signature on a message which the signer (or issuer) must not learn. In particular, the signer must not be able to link a message-signature pair to the execution of the issuing protocol in which it was produced *(blindness)*. Furthermore, it should even for adaptive adversaries be infeasible to produce a valid blind signature without the signing key *(unforgeability)*. Blind signatures have proven to be an important building block for cryptographic protocols, which is featured most prominently in e-cash, e-voting and one-show anonymous credentials.

1.1.3 Anonymous Credentials

Credential systems are means for anonymous authentication and have been foreseen by Chaum as early as in 1985 [Cha85]. Chaum's intention was to develop a protocol that allowed users to interact anonymously with multiple organizations.

[4] Later works on pairing-based schemes often pursue a more abstract approach and use the notion of a bilinear group, which is essentially made up of the pairing, all related groups and their generators. Also in this thesis, we will spare out any technical details related to elliptic curves and only employ the more abstract notion of bilinear groups.

In such a system, a user can obtain a credential from an issuing organization and demonstrate its possession to other organizations acting as verifiers.

This idea was later formalized and extended in [LRSW99] and [CL01] to pseudonym and anonymous credential systems, respectively. While the expressiveness of many early credential systems (e.g., the one in [CL01]) was quite limited, state-of-the-art credential systems typically consider a collection of different attributes (e.g., nationality, sex, etc.). Such systems are known as attribute-based credentials (ABCs) or also as Privacy-ABCs. Here, the credential owner can prove the possession of several attributes in an anonymous fashion to any verifying party.

There are two major lines of anonymous credentials: *one-show* and *multi-show anonymous credentials*. In the former, a user can perform a single unlinkable showing; whereas in the latter a user can conduct an arbitrary number of unlinkable showings. Implementations for one-show and multi-show ABCs are available as Microsoft's U-Prove [BP10] and IBM's idemix [CV02], respectively.

Besides these two types, there are also *delegatable anonymous credentials* [BCC+09]. These allow users to obtain credentials from different organizations and to delegate their credentials to other users later on.

1.1.4 Verifiably Encrypted Signatures

VESs provide means for optimistic fair exchange. A common scenario for this is the following one: Bob wants to buy a theater ticket with an electronic check. That is, he wants to exchange one document, signed by himself, for another document, signed by the theater. If he sends the check before receiving the ticket, he worries that the theater will cash his check without issuing the ticket. On the other hand, the theater is not willing to issue the ticket without receiving a check.

A VES, introduced by Boneh, Gentry, Lynn and Shacham [BGLS03], can be used to resolve this impasse. A VES has two forms of signatures: plain and encrypted. Both forms of signature can be verified, and if the signer refuses to reveal the plain signature at the end of negotiations, the other party can appeal to a trusted third party (called the arbiter), who can recreate a plain signature given the corresponding encrypted signature.

Thus, in our example, the theater can provisionally send Bob a ticket with an encrypted signature, and once they receive Bob's signed check they can reveal the corresponding plain signature, and thus validate the provisional ticket. If they fail to do so, Bob can take the encrypted signature to the arbiter. The arbiter's investigation will reveal that Bob has indeed upheld his side of the deal, and so recreate the corresponding plain signature, giving Bob the ticket he has paid for *(fairness)*. This protocol has the advantage of being *optimistic* meaning that the arbiter need not participate unless there is a dispute.

1.2 This Thesis in a Nutshell

In this thesis, we will not only define a new type of SPSs called structure-preserving signature on equivalence classes (SPS-EQ) but also demonstrate their potential to build cryptographic schemes and protocols in a completely new way. Structure-preserving signatures on equivalence classes (SPS-EQs) allow for a completely new construction paradigm for VESs and privacy-enhancing protocols such as anonymous credentials and blind signatures. The goal that we pursue with SPS-EQ is to define a signature scheme that allows for the consistent and unlinkable randomization of message-signature pairs. To this end, we partition the message space into projective equivalence classes and allow a controlled form of malleability: When signing one class, by signing one of its representatives, we allow subsequent public signature adaptations to arbitrary other representatives. If the decisional Diffie-Hellman (DDH) assumption holds on the underlying group, then we get a form of indistinguishability on the message space for free. Furthermore, in order to obtain security we require a property on the distribution of signatures: After signing a representative, a signer cannot tell apart whether she is being given an adapted signature for a new representative, or a fresh signature on a completely random message. In combination with the property of indistinguishability on the message space, this means that message-signature pairs falling into the same class are indistinguishable.

The consistent and unlinkable randomization of message-signature pairs enables new and highly efficient approaches to especially blind signatures and one-show and multi-show ABCs. A highlight is that for all those schemes blindness and anonymity, respectively, hold against malicious issuers. At its heart, the blinding and unblinding in our blind-signature scheme and one-show ABC scheme are just changes of representatives. Similarly, in the case of our multi-show ABCs, a showing is essentially the presentation of an arbitrary representative of a credential's associated class. Also in our VES scheme we make use of this inherent property of SPS-EQ: The arbiter is able to switch an encrypted signature to its plain form by performing a switch to another representative.

Besides these three main contributions, we introduce a game-based security model for multi-show ABCs, identify flaws in the VES security model and propose a way to fix them. Moreover, we introduce a new commitment type that we require to handle the attribute sets in our multi-show ABC, which is of independent interest. This new primitive allows to commit to sets and to open arbitrary subsets. We present a security model and an efficient construction.

This thesis relies on three publications at major cryptography and computer-security conferences, one preprint and one paper still in review. More precisely, the general results about SPS-EQ stem from [HS14, FHS15a, FHS14] and a full-version paper [FHS16] to [HS14] that is joint work with Georg Fuchsbauer and Daniel Slamanig and has been submitted to Journal of Cryptology. The results on blind signatures and one-show ABCs come from [FHS15a], those on multi-show ABCs from [HS14] and the improvements of these results, which have been incorporated into this thesis, from the respective follow-up paper [FHS16]. Last but not least, the results on VESs are based on [HRS15].

1.3 Related Work

In this section, we discuss work related to the topics SPSs, randomizable signatures, blind and partially blind signatures, set commitments, one- and multi-show ABCs and VESs.

1.3.1 Structure-Preserving Signatures

Digital signatures are an important cryptographic primitive to provide a means for integrity protection, non-repudiation as well as authenticity of messages in a publicly verifiable way. In most signature schemes, the message space consists of integers in $\mathbb{Z}_{|\mathbb{G}|}$ for some group $\mathbb{G}$ or consists of arbitrary strings encoded either to integers in $\mathbb{Z}_{|\mathbb{G}|}$ or to elements of a group $\mathbb{G}$ using a suitable hash function. In the latter case, the hash function is usually required to be modeled as a random oracle (RO) (thus, one signs random group elements).

In contrast, SPSs [Fuc09, AHO10, AFG+10, AGHO11, CDH12, AGOT14b, LPY15, KPW15] can handle messages which are elements of two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ equipped with a bilinear map, without requiring any prior encoding. They have originally been introduced in the context of the GS proof system [GS08]; as new type of signature schemes compatible with this particular proof system defined on bilinear groups [AFG+10]. Later on, other applications of SPS, which are of independent interest, have been found as well [LPJY13, HS14, FHS15a]. Basically, in an SPS scheme the public key, the messages and the signature consist only of group elements and the verification algorithm evaluates a signature by deciding group membership of elements in the signature and by evaluating PPEs. Recently, this concept has been extended to so called fully SPS schemes [AKOT15, Gro15], whose secret keys also only consist of group elements. SPS schemes typically allow to sign vectors of group elements (from one of the two groups $\mathbb{G}_1$ and $\mathbb{G}_2$, or mixed) and also support some types of *randomization* (inner, sequential, etc., cf. [AFG+10, AGOT14b]).

In [AGHO11], Abe et al. showed that Type-3 SPSs (that is, an SPS built over Type-3 bilinear groups) having constant-size signatures cannot exist, unless the signature has at least 3 elements, its elements stem from both groups *(bilateral)* and the SPS scheme uses at least 2 PPEs for verification. In [AGOT14a], they showed similar results for Type-2 SPSs. Moreover, in [AGO11], Abe et al. proved that the unforgeability of optimally short Type-3 SPS schemes (i.e., with 3-element signatures) cannot be reduced to non-interactive assumptions. This means that we are restricted to proving the unforgeability of such schemes in the generic-group model (GGM) (cf. Section 2.2.3).

1.3.2 Randomizable Signatures

In [BFPV11], Blazy et al. present signatures on randomizable ciphertexts (based on linear encryption [BBS04]) using a variant of Waters signatures [Wat05]. Basically, anyone given a signature on a ciphertext can randomize the ciphertext and adapt the signature accordingly, while maintaining public verifiability

and neither knowing the signing key nor the encrypted message. However, as these signatures only allow to randomize the ciphertexts and not the underlying plaintexts, this approach is not useful for our purposes and also not practically efficient.

Another somewhat related approach is the proofless variant of the Chaum-Pedersen signature [CP93] which is used to build self-blindable certificates by Verheul in [Ver01]. This certificate as well as the initial message can be randomized using the same scalar, preserving the validity of the certificate. This approach works for the construction in [Ver01], but it does not represent a secure signature scheme (as also observed in [Ver01]) due to its homomorphic property and the possibility of efficient existential forgeries.

Linearly homomorphic signatures [BFKW09, CFW12, Fre12] allow to sign any subspace of a vector space by producing a signature for every basis vector with respect to the same identifier. The messages are signed together with a unique identifier that "glues" together the single vectors. These signatures are homomorphic, meaning that given a sequence of scalar and signature pairs $(\beta_i, \sigma_i)_{i\in[\ell]}$ for vectors $\vec{v}_i$, one can publicly compute a signature for the vector $\vec{v} = \sum_{i\in[\ell]} \beta_i \vec{v}_i$. If one was using a unique identifier per signed vector $\vec{v}$, then such signatures would support a functionality similar to the one that we are looking for, i.e., publicly compute signatures for vectors $\vec{v}' = \beta\vec{v}$ (although they are not structure-preserving). Various existing constructions also provide the privacy feature of being strongly/completely context-hiding [ALP12, ALP13], meaning that one cannot tell apart a signature for a vector $\vec{v}$ from a signature to a vector $\vec{v}$ resulting from a homomorphic computation on signatures. Nevertheless, this does not help in our context: If we do not restrict every single signed vector to a unique identifier, the signature schemes are homomorphic, which is not compatible with our goal concerning unforgeability. If we apply this restriction, however, then we are not able to achieve our privacy notion as all signatures can be linked to the initial signature by the unique identifier. The same arguments also apply to structure-preserving linearly homomorphic signatures [LPJY13]. Other classes of homomorphic signatures schemes supporting a richer class of admissible functions (beside linear ones) is also related but does not help us either (cf. [ABC+12, ALP12] for an overview).

We note that the general framework of P-homomorphic signatures [ABC+12, ALP12] is related to our approach in terms of unforgeability and privacy guarantees, but there are no existing instantiations for the functionality that we require, and we find it more natural to use our formalization. Moreover, in [CKLM13], the authors introduce a framework for malleable signatures that allows to derive a signature σ' on a message $m' = T(m)$ for an "allowable" transformation T, when given a signature σ for a message m. This can be considered as a generalization of signature schemes, such as quotable [ABC+12, ALP13] or redactable signatures [SBZ02, JMSW02] with the additional property of being context-hiding. The authors note that for messages being pseudonyms and transformations which transfer one pseudonym into another pseudonym, such malleable signatures can be used to construct anonymous credential sys-

tems. They also demonstrate how to build delegatable anonymous credential systems [BCC+09]. The general construction in [CKLM13], however, relies on malleable zero-knowledge proofs (ZKPs) [CKLM12] and is not practically efficient—even when instantiated with the GS proof system [GS08]. Although the above framework is conceptually totally different from our approach, we note that by viewing our scheme in a different way, it fits their definition of malleable signatures (such that their evaluation algorithm takes only a single message vector with corresponding signature and a single type of allowable transformation). However, firstly, our instantiations are far more efficient than their approach (and, in particular, really practical) and, secondly, [CKLM13] when used to construct ABCs, it focuses only on transformations of single messages (pseudonyms) and does not consider attributes (which is the main focus of our construction).

1.3.3 Blind and Partially Blind Signatures

In over 30 years of research, many different (> 50) blind-signature schemes have been proposed. The spectrum ranges from RSA-based (e.g., [Cha82, CKW05]) over DL-based (e.g., [Oka93, Abe01]) and pairing-based (e.g., [Bol03, BFPV11]) to lattice-based (e.g., [Rüc10]) constructions, as well as constructions from general assumptions (e.g., [JLO97, HKKL07, Fis06]). Two distinguishing features of blind signatures are whether they assume a common-reference string (CRS) (cf. Section 2.2.3) set up by a trusted party to which everyone has access; and the number of rounds in the signing protocol. Schemes which require only one round of interaction (two moves) are called *round-optimal* [Fis06]. Besides improving efficiency, round optimality also directly yields concurrent security (which otherwise has to be dealt with explicitly; e.g., [HKKL07]). There are very efficient round-optimal schemes [Cha83, Bol03, BNPS03] under interactive assumptions (chosen-target one-more RSA inversion and chosen-target computational Diffie-Hellman (CDH), respectively) in the random-oracle model (ROM) (cf. Section 2.2.3) as well as under the interactive LRSW [LRSW99] assumption in the CRS model [GS12]. All these schemes are in the honest-key model where blindness only holds against signers whose keys are generated by the experiment.

Fischlin [Fis06] proposed a generic framework for constructing round-optimal blind signatures in the CRS model with blindness under malicious keys: The signer signs a commitment to the message and the blind signature is a non-interactive zero-knowledge proof (NIZKP) of a signed commitment which opens to the message. Using SPS [AFG+10] and the GS proof system [GS08] instead of general NIZKPs, this framework was efficiently instantiated in [AFG+10]. In [BFPV11, BPV12], Blazy et al. gave alternative approaches to compact round-optimal blind signatures in the CRS model which avoid including a GS proof in the final blind signature. Another round-optimal solution with comparable computational costs was proposed by Seo and Cheon in [SC12] building on work by Meiklejohn et al. [MSF10].

Known impossibility results indicate that the design of round-optimal blind signatures in the standard model (cf. Section 2.2.3) has some limitations. Lindell showed in [Lin03] that concurrently secure (and consequently also round-

optimal) blind signatures are impossible in the standard model when using simulation-based security notions. This can however be bypassed via game-based security notions, as shown by Hazay et al. [HKKL07] for non-round-optimal constructions.

Fischlin and Schröder [FS10] showed that black-box reductions of blind-signature unforgeability to non-interactive assumptions in the standard model are impossible if the scheme has three moves or less, blindness holds statistically (or computationally if unforgeability and blindness are unrelated) and protocol transcripts allow to verify whether the user is able to derive a signature (signature-derivation checks). Existing constructions [GRS+11, GG14] bypass these results by making non-black-box use of the underlying primitives (and preventing signature-derivation checks in [GRS+11]).

Garg et al. [GRS+11] proposed the first round-optimal generic construction in the standard model, which can, however, only be considered as a theoretical feasibility result. Using fully homomorphic encryption, the user encrypts the message sent to the signer who evaluates the signing circuit on the ciphertext. To remove the CRS, they use two-move witness-indistinguishable proofs (ZAPs) to let the parties prove honest behavior; to preserve round-optimality, they include the first fixed round of the ZAP in the signer's public key.

Garg and Gupta [GG14] proposed the first efficient round-optimal blind signature constructions in the standard model. They build on Fischlin's framework using SPSs. To remove a trusted setup, they use a two-CRS NIZKP system based on GS proofs and include the CRSs in the public key while forcing the signer to honestly generate the CRS. Their construction, however, requires complexity leveraging (the reduction for unforgeability needs to solve a subexponential discrete logarithm (DL) instance for every signing query) and is proven secure with respect to non-uniform adversaries. Consequently, communication complexity is in the order of hundreds of KB (even at a 80-bit security level) and the computational costs (not considered by the authors) seem to limit their practical application even more significantly.

1.3.4 One-Show ABCs

One-show credential systems are typically built from blind signatures following the approach from Brands [Bra00], which has been implemented in Microsoft's U-Prove [BP10]. Thereby, blind signatures ensure that no party is able to link the credential issuance to any of its showings, while different showings of the same credential are linkable. In 2013, Baldimtsi et al. [BL13b] showed that with currently known proof techniques the underlying blind-signature scheme by Brands [Bra00] cannot be proven secure. To get around this problem, they propose a generic construction of one-show credentials (in the fashion of Brands; called "Anonymous Credentials Light") secure in the ROM [BL13a]. Their credential system is based on a blind-signature scheme that they term blind signatures with attributes, for which they also give a construction based on a non-round-optimal blind-signature scheme by Abe [Abe01].

1.3.5 Set Commitments

The most well-known approach for commitments to (ordered) sets are Merkle hash trees (MHTs) [Mer88], where for a set S the commitment size is $O(1)$ and the opening to a committed value is of size $O(\log |S|)$. Quite recently, Boneh and Corrigan-Gibbs [BC14] proposed an alternative MHT construction using a novel commitment scheme based on a bivariate polynomial modulo RSA composites. In contrast to MHTs, their construction supports succinct proofs of knowledge (PoKs) of committed values.

Kate et al. [KZG10] introduced polynomial-commitment schemes that allow to commit to polynomials and support (batch) openings of polynomial evaluations. They can be used to commit to ordered sets (by fixing an index set) or to sets by considering committed values to be the roots. Their two constructions are analogues to DL and Pedersen commitments and have $O(1)$-size commitments and openings. Recently, Camenisch et al. [CDHK15] proposed a variant of the Pedersen version in [KZG10]. A related commitment scheme, called knowledge commitment, has been proposed in [Gro10] (and later generalized by Lipmaa in [Lip12]).

Another common commitment type for ordered sets are generalized Pedersen [Ped92] or Fujisaki-Okamoto [FO98] commitments. Both have commitment size $O(1)$, but opening proofs are of size $O(|S|)$. For the sake of completeness, we also mention the notion of vector commitments [CF13], which allow to open specific positions as well as subsequent updates at specific positions (but do not necessarily require hiding).

Zero-knowledge sets [MRK03] are another primitive in this context. They allow to commit to a set and to perform membership and non-membership queries on values without revealing any further information on the set. In [DHS15b], it was shown that zero-knowledge sets imply commitments in a black-box way.

1.3.6 Multi-Show ABCs

Signatures providing randomization features [CL03, CL04, BBS04, PS15b] along with efficient zero-knowledge proofs of knowledge (ZKPoKs) of committed values can be used to generically construct ABC systems. The most prominent approaches based on Σ-protocols are CL credentials [CL03, CL04]. With the advent of GS proofs, which allow (efficient) non-interactive proofs in the CRS model without random oracles, various constructions of so-called delegatable (hierarchical) anonymous credentials [BCC$^+$09, Fuc11] and non-interactive anonymous credentials [BCKL08, ILV11] have been proposed. These provide per definition a non-interactive showing protocol, i.e., the show and verify algorithms do not interact when demonstrating the possession of a credential (also the recent model for conventional ABCs in [CKL$^+$14] only support non-interactive showings). In [Fuc11], Fuchsbauer presented the first delegatable anonymous credential system that also provides a non-interactive delegation protocol based on so-called commuting signatures and verifiable encryption. We note that although such credential systems with non-interactive protocols extend the scope of applications

of anonymous credentials, the most common use-case (i.e., authentication and authorization), essentially relies on interaction (to provide freshness/liveness). We emphasize that our goal is not to construct non-interactive anonymous credentials.

1.3.7 Verifiably Encrypted Signatures

VESs and a first instantiation in the ROM were proposed by Boneh, Gentry, Lynn and Shacham [BGLS03]. After their invention, several instantiations were suggested in the ROM [ZSS03, Rüc09] and in the standard model [LOS+06, RS09, Fuc11]. The security model is treated in [BGLS03] and has been further discussed and extended in [Hes04, RS09, CMSW14]. In [RS09], Rückert and Schröder amend the VES security model by new properties and [CMSW14] points out flaws in the security model. To this end, Calderon et al. show in [CMSW14] that secure VESs can be constructed solely from standard digital signatures (DSs). This means, in particular, that the notion of a VES (as previously defined) is not necessarily related to encryption. To exclude such counterintuitive constructions, Calderon et al. introduced as a first step the notion of *resolution independence*, which prevents discrimination between signatures arising from signers and arbiters by requiring plain and resolved signatures to be identically distributed. Interestingly, all known constructions seem to fulfill this property. They further extend this notion to *resolution duplication*. All VESs satisfying this property can then be related to public-key encryption (PKE), that is, PKE can be black-box constructed from such VESs. This result separates resolution-duplicate VESs from one-way function (OWF), which is especially interesting from a theoretical point of view, since DSs can usually be built from OWFs.

1.4 Contribution

As one core contribution, we introduce SPS-EQs. These are a new type of structure-preserving signatures (SPSs) that allow for signing projective equivalence classes set up on the message space $\mathbb{G}^\ell$ (for some prime-order group $\mathbb{G}$). In particular, a signature on one representative can efficiently and publicly be adapted to any other representative of the same class—allowing the consistent and efficient public randomization of message-signature pairs. The main benefit of this primitive is that we can achieve the unlinkability of different message-signature pairs falling into the same class, by: (1) requiring the DDH assumption to hold on $\mathbb{G}$ (making representatives of the same class—loosely speaking—unlinkable for outsiders and for signers not in control of choosing the discrete logarithms of the initially signed representative); (2) fresh and updated signatures to be identically distributed. Thus, SPS-EQs are especially well-suited to design more complex cryptographic protocols, where a signer is required to issue a signature on values out of his (full) control. We show how to use SPS-EQs as an efficient and versatile building block for verifiably-encrypted signatures (VESs)

and privacy-enhancing technologies including but not restricted to (partially) blind signatures, one-show attribute-based credentials (ABCs) and multi-show ABCs.

This thesis relies on three publications at major cryptography and computer-security conferences, one preprint and one paper still in review. More precisely, the general results about SPS-EQ stem from [HS14, FHS15a, FHS14] and a full-version paper [FHS16] to [HS14] that is joint work with Georg Fuchsbauer and Daniel Slamanig and has been submitted to Journal of Cryptology. The results on blind signatures and one-show ABCs come from [FHS15a, FHS15b], those on multi-show ABCs from [HS14] and improvements of these results, which have been incorporated into this thesis, from the respective follow-up paper [FHS16]. Last but not least, the results on VESs are based on [HRS15].

1.4.1 SPS-EQ

We define SPS-EQs by giving their abstract model and an appropriate security model. To this end, we introduce several properties on the distribution of signatures. The weakest property is called *class-hiding*; it demands that randomized message-signature pairs are indistinguishable from fresh message-signature pairs for outsiders. A stronger property is called *perfect adaptation of signatures*; it requires that adapted and fresh signatures have the same distribution. The strongest property is *perfect adaptation of signatures under malicious keys*; roughly speaking it demands adapted signatures to be uniformly distributed in the space of corresponding valid signatures. Both latter properties together with the DDH on the underlying group imply the former property.

Then, we present a malicious-key perfectly adapting SPS-EQ construction secure in the GGM. We further give the first construction whose security relies on a non-interactive assumption. (The scheme does, however, support only the weaker class-hiding notion.) We then show how any SPS-EQ can be turned into an SPS scheme and under which conditions it is rerandomizable. Moreover, we give a black-box separation of malicious-key perfectly adapting SPS-EQs from non-interactive assumptions by means of a meta-reduction. This result plays off the EUF-CMA security and the indistinguishability on the message space and motivates the definition of a weaker unforgeability notion. Finally, we will take this result into account, present a more suitable notion and discuss for which scenarios the original notion is still relevant.

1.4.2 Blind and Partially Blind Signatures

We propose a new paradigm to constructing blind signatures that we call *commit-randomize-derandomize-open approach*. It follows black-box from malicious-key perfectly adapting SPS-EQs and is the first of its kind that is practically efficient, round-optimal (i.e., two-move) and does not rely on a CRS or on ROs. It yields conceptually simple and compact constructions and does not rely on techniques such as complexity leveraging. The main caveat is, however, that blindness is based on a plausible yet interactive assumption. The resulting blind signatures

are entirely practical in terms of key sizes, signature sizes, exchanged group elements and computational effort (when implemented with known instantiations of SPS-EQ a blind signature consists of only 5 bilinear-group elements).

In our blind-signature scheme, the obtainer assembles a representative of an equivalence class as a commitment to the message and a normalization element *(commit)*. She then blinds this message by changing it to another representative *(randomize)*. The signer, given the blinded message, produces a signature on the respective class. Given this signature, the obtainer adapts it to the original representative (containing the original commitment) without requiring the signing key *(derandomize)*. Due to the normalization element, it is guaranteed that the obtainer can only switch back to exactly the original representative. Finally, the blind signature is the rerandomized (unlinkable) signature for the original representative plus an opening for the commitment *(open)*.

We also provide the first construction of round-optimal partially blind signatures in the standard model, which follow straightforwardly from our blind signatures and are almost equally efficient.

1.4.3 One-Show ABCs

We generalize our round-optimal blind-signature scheme to message vectors. Combined with ZKPoKs this almost directly yields one-show anonymous credentials à la "Anonymous Credentials Light" [BL13a]. In this way, we obtain one-show ABCs secure in the standard model (whereas all previous ones come without security proofs or with proofs in the ROM).

1.4.4 Set Commitments

We propose a novel commitment-scheme type that allows for committing to sets and to open arbitrary subsets such that commitments and openings are of size $O(1)$. We propose an abstract model with corresponding security properties. Furthermore, we give a new and efficient set-commitment construction which we prove secure in this model. It is perfectly hiding, allows to commit to subsets of $\mathbb{Z}_p$ and is represented by a single element of a bilinear group. In particular, it allows to open subsets of committed sets. The way the scheme is built also enables commitment rerandomization, which we, however, do not present as an explicit property of set-commitment schemes. The rerandomization is compatible with the rerandomization of the proposed SPS-EQ scheme (i.e., multiplication with a scalar), does not change the set committed to, but requires only a consistent randomization of the witnesses involved in the subset openings. This is a property that cannot be achieved by existing constructions, when one wants to avoid costly ZKPs of randomization.

1.4.5 Multi-Show ABCs

We describe a new way of building multi-show attribute-based anonymous credentials as an application of SPS-EQ and set commitments. In this way, we

are able to construct the first standard-model multi-show ABC with anonymity holding against malicious organization keys. From another perspective, an SPS-EQ scheme allows to consistently randomize a vector of group elements and its signature. So, it seems natural to use this property to achieve unlinkability during the showings in an ABC system. Moreover, it is natural to use the set commitment to commit to the attributes of the user. Loosely speaking, to issue a credential the issuer signs a message (vector) containing the set commitment (and two additional group elements for technical reasons) and the credential is essentially the message and its corresponding signature. During a showing, a subset of the issued attributes can then be opened. The unlinkability of showings is achieved through the rerandomization properties of both the SPS-EQ signature scheme and the set-commitment scheme, whose rerandomizations are compatible with each other. Furthermore, for technical reasons and to thwart replays of showings, we require a small, constant-size ZKPoK. We emphasize that our approach to construct ABCs is very different from existing approaches, as we do not use ZKPs for selectively disclosing attributes during showings. Consequently, we can achieve for our construction that *the size of credentials as well as bandwidth required for showing of a credential* is *independent of the number of attributes in the credential as well as the ABC system*, i.e., a small, constant number of group elements. This is the first ABC system with this feature (Camenisch et al. [CDHK15] recently proposed an approach with identical asymptotic complexity; cf. Section 7.2.7 for details).

We introduce a game-based security model for ABCs in which we prove our ABC system secure. In particular, our security model is a game-based model in the vein of group signatures [BSZ05]. We note that there are no other comprehensive models available for ABC systems (apart from independently developed very strong simulation-based notions in [CKL+14, CDHK15]). Moreover, we consider replays and a strong form of anonymity against organizations that may generate malicious keys (without any CRS)—both are things that many earlier works and models do not consider. Especially replays have often been considered as an issue that is delegated to the implementation of an ABC system. But we want to prevent such attacks already in the formal analysis of an ABC system to avoid problems that appear later within an implementation that may simply not consider or ignore this issue due to a lack of better knowledge. We note that the recent independent formal model by Camenisch et al. in [CKL+14] and the recent ABC construction by Camenisch et al. in [CDHK15]—using a different model—do consider replays and malicious keys too, although the former in a seemingly weaker sense and the latter only in context of a CRS. As another contribution, we discuss a scheme variant with smaller organization key sizes that is concurrently secure in the CRS model.

Finally, for the sake of completeness, we compare our ABC system to other existing multi- and one-show ABC approaches.

1.4.6 Verifiably Encrypted Signatures

We propose the first black-box construction of a VES from any SPS-EQ satisfying a simple property that we term *perfect composition* and which resembles the resolution duplication property of VESs. This construction does *not* combine an encryption scheme with an SPS-EQ, but allows us to show a connection between perfectly composing SPS-EQs and PKE. Furthermore, all our security proofs hold in the standard model, under the Diffie-Hellman inversion (DHI) assumption.

We also revisit the security definitions of VES. The original definition of VES [BGLS03] requires that the underlying (ordinary) signature scheme be correct and secure in addition to other security properties. The latter properties have been extended in subsequent literature [Hes04, RS09] but the requirements on the underlying scheme are sometimes neglected. We show that with this omission, resolution independence is absolutely essential not only to the unforgeability, but even to the correctness, of the underlying signature scheme. From the alternative viewpoint, we show that security including resolution independence is sufficient for the correctness and security of the underlying signature scheme.

Public-Key Encryption

We propose the first black-box construction of a CPA-secure PKE scheme from any SPS-EQ allowing perfect composition. The construction follows the idea of Calderon et al. [CMSW14]; it is black-box and does not involve known non-black-box techniques such as zero knowledge (ZK). Given the well-known impossibility results, this shows that SPS-EQs allowing perfect composition cannot be constructed from OWFs in a black-box way.

1.5 Other Contributions

Other papers published during the PhD studies, but not incorporated into this thesis, deal with revocable multi-show ABCs [DHS15a], cryptographic accumulators [DHS15b], provable data possession [HS13b], proxy signatures [HS13c, HS13a, DHS14b, DHPS15] and cryptographic software implementations including but not limited to implementations of ECC [PWH+13, HW13, GH13b, DHS14a]. Last but not least, during the time of his PhD studies, the author has developed and maintained large parts of the versatile, commercial Java™ ECC library ECCelerate™ [HR], which offers amongst others support for bilinear pairings.

1.6 Structure of this Thesis

Chapter 2 discusses the preliminaries and general definitions required for this thesis. In Chapter 3, we introduce structure-preserving signatures on equiva-

lence classes (SPS-EQs), which we then use to build round-optimal (partially) blind signatures in Chapter 4 and verifiably encrypted signatures in Chapter 5. In Chapter 6, we introduce set commitments and in Chapter 7, we construct multi-show ABCs from SPS-EQ and set commitments and one-show ABCs from our blind-signature construction, respectively. Finally, Chapter 8 concludes this thesis and discusses open issues.

2 Preliminaries

They split the cryptographic key.
— Stephen Levy, on Diffie and Hellman

A hard problem is one that nobody works on.
— James L. Massey

We start with some preliminary definitions which will recur throughout the whole thesis. Next, we will discuss some complexity-theoretic background and we will continue by considering bilinear groups and complexity assumptions in this context. Finally, we will briefly introduce cryptographic primitives; among others, we will discuss commitments, proofs of knowledge (PoKs), digital signatures (DSs) and public-key encryption (PKE).

The following stems to some extent from [FHS15a, FHS15b, DHS15b, HRS15, FHS16]; other parts are based on [Kat10, KL15, FS10, Dam10, Sch15]. We often do not provide explicit references.

2.1 General Definitions and Notation

A function $\epsilon\colon \mathbb{N} \to \mathbb{R}^+$ is called *negligible* if $\forall\, c > 0\ \exists\, k_0\ \forall\, k > k_0 : \epsilon(k) < 1/k^c$. By $a \xleftarrow{R} S$ we denote that a is chosen uniformly at random from a set S.

We call an algorithm $\mathsf{A}(a_1, \ldots, a_n)$ *algebraic* with respect to an algebraic structure S (e.g., a group, ring or field), if it gets input the encoding of elements of S and outputs encodings of elements of S obtained by processing the input using only algebraic manipulations defined on S. An algorithm $\mathsf{A}(a_1, \ldots, a_n)$ is said

to be *probabilistic-polynomial time (PPT)* if it internally uses randomness and its running time is polynomially bounded in its input size. We write $\mathsf{A}(a_1, \ldots, a_n; r)$ to make the randomness r used by a PPT algorithm $\mathsf{A}(a_1, \ldots, a_n)$ explicit.

Furthermore, we say that a function $f\colon \{0,1\}^* \to \{0,1\}^*$ is *efficiently computable* or *easy to compute*, if it can be computed by a PPT algorithm, that is, there exists a PPT algorithm A_f such that $\mathsf{A}_f(x) = f(x)$ for all $x \in \{0,1\}^*$.

With the notation $\Pr[\,\mathbf{Exp} : E\,]$, we denote the probability of a particular event E occurring after the execution of a random experiment **Exp**. Experiment **Exp** is a sequence of operations (and algorithm invocations), which are sequentially executed from left to right; Event E is typically represented by a predicate and occurs if the predicate evaluates to 1.

2.1.1 One-Way Functions

A *one-way function (OWF)* is a function that is efficiently computable, but computationally hard to invert when given its evaluation on random values:

Definition 2.1 (One-way function)**.** *A function $f\colon \{0,1\}^* \to \{0,1\}^*$ is* one-way*, if f is efficiently computable and for all PPT adversaries $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that:*

$$\Pr\left[\, x \xleftarrow{R} \{0,1\}^\kappa,\ x' \xleftarrow{R} \mathcal{A}(1^\kappa, f, f(x)) \ :\ f(x) = f(x')\,\right] \leq \epsilon(\kappa).$$

A *trapdoor function* or *trapdoor one-way function* $f\colon \{0,1\}^* \to \{0,1\}^*$ is a one-way function, which becomes easy to invert when given some additional piece of information—the *trapdoor*.

OWFs are fundamental to cryptography, yet their existence is still unknown. In their seminal work [IR89], Impagliazzo and Rudich showed that cryptographic primitives can be classified as lying in one of two "worlds". The "Minicrypt" world contains those primitives which are equivalent to the weakest known assumption—the existence of OWFs. This includes, for instance, digital signature (DS) schemes (cf., e.g., [DH76, RSA78, GMR88]), pseudo-random generators [Lam79, IL89, GL89, Rom90, HILL99] and, in further consequence also pseudo-random functions [GGM84]. The second world, called "Cryptomania", includes primitives that require stronger assumptions such as PKE [RSA78] or key agreement (KA) [DH76].

2.1.2 Hard-Core Predicates

Even though, OWFs are allegedly computationally hard to invert, partial information about x can still leak when given $f(x)$. In particular, only recovering x in its entirety is assumed to be infeasible for all PPT algorithms. A hard-core predicate of a one-way function is a function with range $\{0,1\}$, which is easy to compute when given x, but hard to compute with probability significantly better than $1/2$, when given $f(x)$. In some sense, hard-core predicates allow to capture the difficulty of inverting f.

Definition 2.2 (Hard-core predicate)**.** *A* hard-core predicate $b\colon \{0,1\}^* \to \{0,1\}$ *of a one-way function* $f\colon \{0,1\}^* \to \{0,1\}^*$ *is a function such that for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that*

$$\Pr\left[\, x \xleftarrow{R} \{0,1\}^\kappa,\ b^* \xleftarrow{R} \mathcal{A}(1^\kappa, f, f(x), b) \ :\ b^* = b(x) \,\right] - \frac{1}{2} \leq \epsilon(\kappa).$$

In [GL89], Goldreich and Levin showed how to construct a hard-core predicate from any OWF f: Let $b(x,r) \coloneqq \langle x, r\rangle$ with $|r| = |x|$ and $\langle\cdot,\cdot\rangle$ being the inner product on the vector space $\mathbb{Z}_2^{|x|}$, then b is a hard-core predicate for f.

2.1.3 Hash Functions

An *(unkeyed) hash function* $h_\kappa\colon \{0,1\}^* \to \{0,1\}^\kappa$ is a deterministic polynomial-time algorithm mapping an arbitrary-length message to a bitstring of fixed size κ, the *hash value* or *message digest*. Typically, we assume a hash function to be collision resistant [Dam88], that is, it should be computationally hard to find two values $x, y \in \{0,1\}^*$ yielding the same hash value, i.e., $h_\kappa(x) = h_\kappa(y)$. Thereby, collision resistance for an unkeyed hash function can only hold asymptotically in front of uniform adversaries [Rog06]:

Definition 2.3 (Collision-resistant hash function)**.** *Let* $h_\kappa\colon \{0,1\}^* \to \{0,1\}^\kappa$*, then* h_κ *is a* collision-resistant hash function *if* h_κ *is efficiently computable and for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\, (x, x') \xleftarrow{R} \mathcal{A}(1^\kappa, h_\kappa) \ :\ h_\kappa(x) = h_\kappa(x') \,\right] \leq \epsilon(\kappa).$$

In particular, defining collision resistance for unkeyed hash functions is problematic in theory, since there is always a constant-time algorithm that outputs a collision (x, x'): The algorithm that contains a collision in hard-coded form. This is circumvented by the asymptotic definition, as it is impossible to hardwire a collision (x, x') for every possible security parameter $\kappa > 0$. Likewise, keyed hash functions, that is $h\colon K \times \{0,1\}^* \to \{0,1\}^\kappa$ (with K being the key space of size exponential in κ; and keys $k \in K$ to instantiate a concrete hash function made public), allow to bypass this problem for a similar reason: It is impossible to hardwire a collision (x, x') for every key $k \in K$ without using an unreasonable portion of space.

Hash functions used in practice are usually unkeyed and have a fixed output length. Thus, they do not satisfy any theoretical definition of collision resistance. When we consider such a hash function as being collision-resistant, this does not mean that there is no simple adversary against this property, but just that no such adversary is known to us humans [Rog06].

Other notions of security are *preimage resistance*, which essentially requires h_κ to be one-way, and *second-preimage resistance*, which demands finding a second preimage $x' \neq x$ being infeasible when given h_κ and a uniform preimage x. The latter property is implied by collision resistance.

2.2 Some Background on Cryptographic Complexity Theory

We briefly discuss the most important complexity-theoretic topics that are required throughout this thesis.

2.2.1 Hard Problems

In cryptography, we are particularly interested in proving the security of cryptographic primitives. There are several ways to prove security: In some cases it can be shown in a perfect (information-theoretic) way (e.g., for the one-time pad or the hiding property of Pedersen commitments, cf., Section 2.5.1); in most other cases we have to base the primitives' security on problems which we believe to be hard to solve. Breaking the primitive allows us to solve the underlying problem contradicting the hardness assumption.

There are two major types of hard problems: *non-interactive* and *interactive* problems. Non-interactive problems split up into *parameterized* (*q-type*) and *static* (or *unparameterized*) problems, where the size of the problem instance in the former case depends on a parameter q that itself depends on the adversarial behavior. Interactive problems form the weakest class of problems; here an adversary interacts with the instance generator during the generation of the problem instance. Overall, *static* problems are the most desirable class of problems.

Another important characterization is *falsifiability* [Nao03, GW11, GK16]. A problem P is falsifiable, if it can always be efficiently verified whether an adversary $\mathcal{A}$ is successful in breaking it, i.e., whether a value x output by $\mathcal{A}$, when being run on instance y of problem P, is a valid solution to y or not. In particular, there are two types of such problems: Problems that allow the efficient public verification of a solution x to an instance y and problems that only allow this for the instance generator (or, alternatively, if the randomness r used during the generation of y is known to the verifier). In fact, non-interactive problems are falsifiable [Nao03, GK16]. For interactive problems to be somewhat plausible, we require them to be at least falsifiable.

We will now give a definition of a non-interactive problem and its hardness.

Definition 2.4 (Non-interactive problem)**.** *A* non-interactive problem P *consists of the following PPT algorithms:*

$\mathsf{IGen}(1^\kappa; r)$: *A probabilistic algorithm that takes input a security parameter* 1^κ *(and has access to a random tape* $r \in \{0,1\}^*$*). It outputs an instance* y *of* P*.*

$\mathsf{V}(x, y, r)$: *A deterministic algorithm that takes input a value* x*, an instance* y *of* P *and randomness* $r \in \{0,1\}^*$ *such that* y *was generated using* r*. It outputs a decision bit indicating whether or not* x *is a solution of* y*.*

Definition 2.5 (Hard non-interactive problem)**.** *A non-interactive problem* P *is* hard*, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\, y \xleftarrow{R} \mathsf{IGen}(1^\kappa; r),\ x \xleftarrow{R} \mathcal{A}(y)\ :\ \mathsf{V}(x, y, r) = 1 \,\right] \leq \epsilon(\kappa).$$

2.2.2 Black-Box Relations and Separations

We say that an algorithm $\mathcal{R}$ has *black-box access* or *oracle access* to another algorithm $\mathcal{A}$ (denoted by $\mathcal{R}^{\mathcal{A}}$), if $\mathcal{R}$ and $\mathcal{A}$ are given the same security parameter and $\mathcal{R}$ is allowed to query $\mathcal{A}$ as an oracle an arbitrary number of times in an interleaving fashion. If $\mathcal{A}$ is probabilistic, then all copies of $\mathcal{A}$ use the same random tape, which $\mathcal{R}$, however, cannot access.

Loosely speaking, a *black-box separation* is an indication for the hardness of finding a reduction between two primitives C_1 and C_2. It allows to identify a gap between them (i.e., separating C_1 from C_2), by ruling out the existence of a reduction between them.

Oracle separations are one strategy to this end. They go back to a seminal paper by Impagliazzo and Rudich [IR90]. Among other separation results [Rud92, Sim98, KST99, GKM+00, GT00, GMR01, Fis02, FS10], they also allow to divide cryptographic primitives into different realms (as already discussed in Section 2.1.1).

Another methodology for black-box separations are meta-reductions [BV98, Cor02, PV06, BMV08], which are reductions against reductions. They allow us to show that the pure existence of a reduction $\mathcal{R}$ from a primitive C to a hard problem already violates some hardness assumption P. The starting point here is a (potentially inefficient) forger $\mathcal{F}$ to which a reduction $\mathcal{R}$ is given black-box access. The goal is then to construct a meta-reduction $\mathcal{M}$ that when having access to $\mathcal{R}$ can efficiently simulate the environment (including forger $\mathcal{F}$) for $\mathcal{R}$ and use the output of $\mathcal{R}$ to solve P.

2.2.3 Computational Models

There are different computational models, in which we are able to conduct security proofs; most prominently, the *random-oracle model (ROM)*, the *common-reference string (CRS) model* and the *standard (or plain) model*. We will also discuss the *generic-group model (GGM)*, as we will refer to it later on.

The Random-Oracle Model

In the ROM [BR93], hash functions are idealized and modeled via oracles, which means that an adversary $\mathcal{A}$ is unable to evaluate a hash function $h\colon \{0,1\}^* \to \{0,1\}^\kappa$ by itself. Each time $\mathcal{A}$ evaluates h, $\mathcal{A}$ must query to a hash oracle $\mathcal{O}^h\colon$ $\mathcal{O}^h$ on input m responds with a uniformly random string of length κ if m has not been queried before; and with the same answer as in the first query for m otherwise. There are some (artificial) pathological cases of ROM-secure schemes, which turn out being insecure when plugging in any real hash function [CGH98, GR04]. The

bottom line is that the ROM only gives a heuristical indication for the security of a scheme.

The Generic-Group Model

The GGM [Sho97, Mau05] is an idealized cryptographic model to abstract algebraic computations performed in a group. Thereby, an adversary is forced to access group-action oracles in order to perform computations in the group. These oracles return an encoding of resulting group elements. As in the ROM, the encodings are chosen uniformly at random and encodings for already queried elements are being returned consistently to previous queries. Similar to the ROM, a GGM analysis is only a necessary condition for security and only gives a heuristical indication that no algebraic adversary is able to break a certain primitive: It suffers from similar shortcomings as the ROM [CGH98, Den02].

The Common-Reference-String Model

The CRS model (also known as *common-random-string* or as *auxiliary-string* model) and also other variations like the registered public-key model or the bare public-key model assume that a trusted third party (TTP) correctly performs a trusted setup and outputs a CRS (or equivalently public parameters), to which all participating parties are given access. After the setup the TTP is no longer available. However, since the TTP is overly powerful, a trusted setup is by itself a strong assumption.

The Standard Model

In contrast to aforementioned models, the standard model does not provide any additional aids. The security of schemes with proofs in the standard model relies solely on the used complexity assumptions. Thus, it is the strongest and most desirable model.

2.3 Bilinear Groups

A bilinear map (or pairing) maps elements stemming from two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ to a target group $\mathbb{G}_T$. The former two groups are related in the sense that they are defined by the same elliptic curve equation (over a base field $\mathbb{F}$). The latter is a multiplicative subgroup of an extension field of $\mathbb{F}$.

More precisely, a bilinear map is defined as follows:

Definition 2.6 (Bilinear map)**.** *Let* $(\mathbb{G}_1, +)$*,* $(\mathbb{G}_2, +)$*, generated by* P *and* $\hat{P}$*, respectively, and* $(\mathbb{G}_T, \cdot)$ *be cyclic groups of prime order* p*. We call* $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ *a* bilinear map *or* pairing *if it is efficiently computable and it holds that:*

Bilinearity: $e(aP, b\hat{P}) = e(P, \hat{P})^{ab} = e(bP, a\hat{P}) \quad \forall\, a, b \in \mathbb{Z}_p$*.*

Non-degeneracy: $e(P, \hat{P}) \neq 1_{\mathbb{G}_T}$*, i.e.,* $e(P, \hat{P})$ *generates* $\mathbb{G}_T$*.*

If $\mathbb{G}_1 = \mathbb{G}_2$, then e is said to be a *symmetric* (or Type-1) pairing and *asymmetric* otherwise. In the latter case, we distinguish between *Type-2* and *Type-3 pairings*: For Type-2 pairings an efficiently computable isomorphism $\Psi\colon \mathbb{G}_2 \to \mathbb{G}_1$ is known, whereas for Type-3 pairings such a map is not known to exist. Type-3 pairings are currently the most efficient choice with regard to a security/efficiency trade-off [CM11].

For our purposes, we will mostly use Type-3 pairings and, moreover, only consider groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ having the same prime order p.[1] In particular, if $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = p$, we say that the bilinear group BG has order p. We will consider bilinear maps in a rather abstract way and, therefore, summarize such pairings and all related entities in a so-called bilinear-group description $\mathsf{BG}_3 = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$, for whose generation, we introduce the following algorithm:

Definition 2.7 (Type-3 bilinear-group generator)**.** *A* bilinear-group generator BGGen_3 *is a PPT algorithm that takes a security parameter* 1^κ *and outputs a bilinear group* $\mathsf{BG}_3 = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$ *consisting of three prime-order* p *groups* $\mathbb{G}_1 = \langle P \rangle$, $\mathbb{G}_2 = \langle \hat{P} \rangle$ *and* $\mathbb{G}_T$ *with* $\log_2 p = \lceil \kappa \rceil$ *and a pairing* $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

In contrast, a Type-1 bilinear group BG_1 has the form $\mathsf{BG}_1 = (p, \mathbb{G}, \mathbb{G}_T, e, P)$; whereas a Type-2 bilinear group additionally takes the efficiently computable isomorphism $\Psi\colon \mathbb{G}_2 \to \mathbb{G}_1$ into account, that is, $\mathsf{BG}_2 = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P}, \Psi)$.

Subsequently, we will focus on Type-3 bilinear groups. For the sake of simplicity, we will write BG and BGGen for BG_3 and BGGen_3, respectively. Moreover, for technical reasons we will sometimes assume BGGen to be deterministic. This is, e.g., the case for Barreto-Naehrig curves—the most common choice for Type-3 pairings [BN06].

2.3.1 Complexity Assumptions

Now, we will discuss some common complexity assumptions in the Type-3 bilinear group context, which we are going to use throughout this thesis.

Definition 2.8 (Discrete logarithm (DL) assumption)**.** *Let* BGGen *be a bilinear-group generator that outputs* $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1 = P, P_2 = \hat{P})$*. Then, the* DL assumption *holds for* BGGen *in* $\mathbb{G}_i$ *if for every PPT adversary* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\, \mathsf{BG} \xleftarrow{R} \mathsf{BGGen}(1^\kappa),\ a \xleftarrow{R} \mathbb{Z}_p,\ a' \xleftarrow{R} \mathcal{A}(\mathsf{BG}, aP_i) \ :\ a'P_i = aP_i \,\right] \leq \epsilon(\kappa).$$

Note that the discrete logarithm (DL) assumption is the weakest assumption (or the hardest problem) in (bilinear-)group-based cryptography: An efficient solver for it enables us to solve any other problem in this setting.

Another important computational standard assumption is the following one:

[1] Note that composite-order bilinear groups exist as well [BGN05].

Definition 2.9 (Computational Diffie-Hellman (CDH) assumption)**.** *Let* BGGen *be a bilinear-group generator that outputs* $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1 = P, P_2 = \hat{P})$*. Then, the* CDH assumption *holds for* BGGen *in* $\mathbb{G}_i$ *if for every PPT adversary* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\, \mathsf{BG} \xleftarrow{R} \mathsf{BGGen}(1^\kappa),\ r, s \xleftarrow{R} \mathbb{Z}_p,\ T_i \xleftarrow{R} \mathcal{A}(\mathsf{BG}, rP_i, sP_i) \ :\ T_i = rsP_i \,\right] \leq \epsilon(\kappa).$$

The Diffie-Hellman inversion (DHI) assumption [MSK02], which we consider next turns out useful in bilinear groups and is equivalent to the computational Diffie-Hellman (CDH) assumption [BDZ03].

Definition 2.10 (Diffie-Hellman inversion (DHI) assumption)**.** *Let* BGGen *be a bilinear-group generator that outputs* $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1 = P, P_2 = \hat{P})$*. Then, the* DHI assumption *holds for* BGGen *in* $\mathbb{G}_i$ *if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\, \mathsf{BG} \xleftarrow{R} \mathsf{BGGen}(1^\kappa),\ a \xleftarrow{R} \mathbb{Z}_p^*,\ T_i \xleftarrow{R} \mathcal{A}(\mathsf{BG}, aP_i) \ :\ T_i = \frac{1}{a} P_i \,\right] \leq \epsilon(\kappa).$$

The co-DHI assumption is similar, yet stronger, as it makes the instance value available in both groups $\mathbb{G}_1$ and $\mathbb{G}_2$ [CM11, FHS15a] (and, thus, implies the above assumption):

Definition 2.11 (co-Diffie-Hellman inversion (co-DHI) assumption)**.** *Let* BGGen *be a bilinear-group generator that outputs* $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$*. Then, the* co-DHI assumption *holds for* BGGen *in* $\mathbb{G}_1$ *if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l|r} \mathsf{BG} \xleftarrow{R} \mathsf{BGGen}(1^\kappa),\ a \xleftarrow{R} \mathbb{Z}_p^*, & T \in \mathbb{G}_1 \\ T \xleftarrow{R} \mathcal{A}(\mathsf{BG}, aP, a\hat{P}) \quad : & \wedge\ e(T, a\hat{P}) = e(P, \hat{P}) \end{array}\right] \leq \epsilon(\kappa).$$

The co-DHI assumption can be defined analogously for $\mathbb{G}_2$; with the difference that we require $\mathcal{A}$ to output $\hat{T} \in \mathbb{G}_2$. This assumption is implied by a version of the decision linear (DLIN) assumption [BBS04] in Type-3 bilinear groups, which demands that it is hard to distinguish $T = (r + s)P_1$ from $R \xleftarrow{R} \mathbb{G}_1$ when given $(\mathsf{BG}, (aP_j, bP_j)_{j\in[2]}, raP_2, sbP_2)$, where $P_1 = P$, $P_2 = \hat{P}$ and $a, b, r, s \xleftarrow{R} \mathbb{Z}_p^*$.[2]

We introduce the following assumption, which is, e.g., implied by the Type-3 bilinear-group counterpart of the q-co-DHI assumption [BDZ03, CM11]—a generalization of the co-DHI assumption having problem instance $(a^jP, a^j\hat{P})_{j\in[q]}$; or, e.g., by the q-co-strong Diffie-Hellman (SDH) assumption [BB04, CM11] in $\mathbb{G}_i$, which has problem instance $(a^jP, a^j\hat{P})_{j\in[q]}$ and requires $\mathcal{A}$ to output $(c, \frac{1}{a+c}P_i)$ with $c \in \mathbb{Z}_p \setminus \{-a\}$:

[2] To see this, observe that a solver for co-DHI instances can be used to compute $\frac{1}{a}P$ and $\frac{1}{b}P$, which then allows to check whether $e(\frac{1}{a}P, ra\hat{P}) \cdot e(\frac{1}{b}P, sb\hat{P}) = e(T, \hat{P})$. The same holds for the co-DHI assumption in $\mathbb{G}_2$.

Definition 2.12 (q-co-Discrete logarithm (q-co-DL) assumption)**.** *Let* BGGen *be a bilinear-group generator that outputs* $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$*. Then, the* q-co-DL assumption *holds for* BGGen*, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l}\mathsf{BG} \xleftarrow{R} \mathsf{BGGen}(1^\kappa),\ a \xleftarrow{R} \mathbb{Z}_p, \\ a' \xleftarrow{R} \mathcal{A}(\mathsf{BG}, (a^j P, a^j \hat{P})_{j\in[q]})\end{array} \ :\ a'P = aP\right] \leq \epsilon(\kappa).$$

Note that we will use the q-co-DL assumption statically throughout this thesis, that is, q is a fixed system parameter and does not depend on the adversary's behavior, as, e.g., in [BB04].

The decisional counterpart of the CDH assumption (straightforwardly implying the latter) is defined as follows:

Definition 2.13 (Decisional Diffie-Hellman (DDH) assumption)**.** *Let* BGGen *be a bilinear-group generator that outputs* $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1 = P, P_2 = \hat{P})$*. Then, the* decisional Diffie-Hellman (DDH) assumption *holds for* BGGen *in* $\mathbb{G}_i$ *if for every PPT adversary* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l}b \xleftarrow{R} \{0,1\},\ \mathsf{BG} \xleftarrow{R} \mathsf{BGGen}(1^\kappa),\ r,s,t \xleftarrow{R} \mathbb{Z}_p, \\ b^* \xleftarrow{R} \mathcal{A}(\mathsf{BG}, rP_i, sP_i, (b\cdot rs + (1-b)\cdot t)P_i)\end{array} \ :\ b^* = b\right] - \frac{1}{2} \leq \epsilon(\kappa).$$

The *eXternal Diffie-Hellman (XDH) assumption* implies the DDH to hold in $\mathbb{G}_1$ and, in doing so, formalizes the alleged absence of efficiently computable isomorphisms from $\mathbb{G}_1$ to $\mathbb{G}_2$ in Type-2 bilinear groups. Likewise, the *Symmetric eXternal Diffie-Hellman (SXDH) assumption* implies the DDH to hold in both $\mathbb{G}_1$ and $\mathbb{G}_2$ and, thus, formalizes the assumption that in Type-3 bilinear groups there is no efficiently computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ either: [3]

Definition 2.14 (Symmetric eXternal Diffie-Hellman (SXDH))**.** *The* SXDH assumption *holds for* BGGen *if the DDH problem holds in both* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, respectively.*

The Generalized co-SDH Assumption

The last assumption we use (Definition 2.16) falls into the uber-assumption family [Boy08, Corollary 1] for the Type-3 bilinear-group setting, which we state for completeness:

Definition 2.15 ($(\mathsf{R}, \mathsf{S}, \mathsf{T}, f)$-Diffie-Hellman assumption)**.** *Let* BGGen *be a bilinear-group generator that outputs* $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$*; further let* $\mathbf{g} \leftarrow e(P, \hat{P})$ *and* $\mathsf{R} = (\mathsf{r}_i)_{i\in[r]}, \mathsf{S} = (\mathsf{s}_j)_{j\in[s]}, \mathsf{T} = (\mathsf{t}_k)_{k\in[t]}$ *be three tuples of multivariate polynomials in* $\mathbb{Z}_p[X_1, \ldots, X_n]$*. Define* $\mathsf{R}(\vec{x}) := (\mathsf{r}_i(\vec{x})P)_{i\in[r]}$*,* $\mathsf{S}(\vec{x}) := (\mathsf{s}_i(\vec{x})\hat{P})_{i\in[s]}$ *and* $\mathsf{T}(\vec{x}) := (\mathbf{g}^{\mathsf{t}_i(\vec{x})})_{i\in[t]}$*. Then, the* $(\mathsf{R}, \mathsf{S}, \mathsf{T}, f)$*-DH assumption*

[3] Note that for Type-1 bilinear groups, a pairing $e\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ allows to efficiently decide the DDH problem: Given a DDH instance $(\mathsf{BG}_1, aP, bP, cP)$, check whether $e(aP, bP) = e(cP, P)$. Likewise, for Type-2 bilinear groups the DDH is efficiently decidable in $\mathbb{G}_2$: Given $(\mathsf{BG}_2, a\hat{P}, b\hat{P}, c\hat{P})$, check whether $e(\Psi(a\hat{P}), b\hat{P}) = e(P, c\hat{P})$.

holds for BGGen*, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l} \mathsf{BG} \xleftarrow{R} \mathsf{BGGen}(1^\kappa),\ \vec{x} \xleftarrow{R} \mathbb{Z}_p^n, \\ \mathbf{g}^{f(\vec{x})} \xleftarrow{R} \mathcal{A}(\mathsf{BG}, \mathsf{R}(\vec{x}), \mathsf{S}(\vec{x}), \mathsf{T}(\vec{x})) \end{array} : \begin{array}{r} 0 \neq f \in \mathbb{Z}_p[X_1, \ldots, X_n] \\ \wedge\ f \neq \sum_{(i,j)\in[r]\times[s]} A_{ij}\mathsf{r}_i\mathsf{s}_j + \\ + \sum_{k\in[t]} b_k \mathsf{t}_k \\ \forall\, A \in \mathbb{Z}_p^{r\times s}\ \forall\, \vec{b} \in \mathbb{Z}_p^t \end{array}\right] \leq \epsilon(\kappa).$$

Essentially, this assumption says that it is hard to evaluate a polynomial $f \in \mathbb{Z}_p[X_1, ..., X_n]$ at vector $\vec{x} \in \mathbb{Z}_p^n$ such that f is independent of the polynomials in R, S and T, whose evaluations at $\vec{x}$ are given to $\mathcal{A}$.

We introduce the following assumption, which is implied by the above assumption and generalizes the q-co-SDH assumption from [BB04, CM11]. The latter states that given $(a^i P, a^i \hat{P})_{i\in[q]}$, it is hard to output $(s, \frac{1}{a+s}P)$ for any s. This can be interpreted as outputting the polynomial $h(X) := X + s$ and $\frac{1}{h(a)}P$. The next assumption states that it is not only hard to compute $\frac{1}{h(a)}P$ for h of that specific form, but it is also hard to compute $\frac{g(a)}{h(a)}P$ for any non-zero polynomials g, h for which $\deg g < \deg h$.

Definition 2.16 (Generalized q-co-SDH assumption)**.** *Let* BGGen *be a bilinear-group generator that outputs* $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$*. Then, the generalized* q*-co-SDH assumption holds for* BGGen *in* $\mathbb{G}_1$*, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l} \mathsf{BG} \xleftarrow{R} \mathsf{BGGen}(1^\kappa),\ a \xleftarrow{R} \mathbb{Z}_p, \\ \big(g, h, T\big) \xleftarrow{R} \\ \qquad \mathcal{A}(\mathsf{BG}, (a^i P, a^i \hat{P})_{i\in[q]}) \end{array} : \begin{array}{r} T \in \mathbb{G}_1\ \wedge\ g, h \in \mathbb{Z}_p[X] \\ \wedge\ 0 \leq \deg g < \deg h \leq q \\ \wedge\ e(T, h(a)\hat{P}) = e(g(a)P, \hat{P}) \end{array}\right] \leq \epsilon(\kappa).$$

Analogously, the above assumption can be defined to require $T \in \mathbb{G}_2$. As with the q-co-DL assumption, we will use the generalized q-co-SDH assumption statically.

It allows exponentially many solutions and involves rational polynomials. Thus, to cover it with the uber-assumption framework[4], we introduce the following family of rational target polynomials $\mathcal{F}_q = \{\frac{g(X)}{h(X)} : g, h \in \mathbb{Z}_p[X], 0 \leq \deg g < \deg h \leq q\}$. Then, we require the adversary to additionally specify the target polynomial $f \in \mathcal{F}_q$. It can easily be seen that for $(\mathsf{R}, \mathsf{S}, \mathsf{T}) = ((X^i)_{i\in[0,q]}, (X^i)_{i\in[0,q]}, 1)$ and any $f \in \mathcal{F}_q$ the generalized q-co-SDH assumption is implied by the $(\mathsf{R}, \mathsf{S}, \mathsf{T}, f)$-Diffie Hellman assumption: Observe that the generalized q-co-SDH assumption demands the solution to be in $\mathbb{G}_1$ and that any $f = \frac{g}{h} \in \mathcal{F}_q$ is—due to being rational—independent from all polynomials in $\mathsf{R}, \mathsf{S}, \mathsf{T}$. The asymptotic simulation error in the GGM proof of the generalized q-co-SDH assumption attains a cubic error bound.

[4] As generally discussed and, in particular, demonstrated for, e.g., the similar but weaker SDH assumption in [Boy08, Sections 6.1 and 6.2]: There, the target polynomial f is allowed to be rational and a family $\mathcal{F} = \{\frac{1}{h(X)} : h \in \mathbb{Z}_p[X], \deg h = 1\}$ is used to describe all possible target polynomials (as there are exponentially many). It must be particularly taken care of that its denominator does not vanish.

2.4 Public-Key Encryption

We now give the abstract model of a PKE scheme and basic security definitions.

Definition 2.17 (Public-key encryption (PKE) scheme)**.** *A* PKE scheme PKE *consists of the following PPT algorithms.*

$\mathsf{KeyGen}(1^\kappa)$: *A probabilistic algorithm that takes input a security parameter* 1^κ*. It returns a key pair* $(\mathsf{sk}, \mathsf{pk})$ *for plaintext space* $\mathcal{M}_{\mathsf{pk}}$ *and ciphertext space* $\mathcal{C}_{\mathsf{pk}}$*.*

$\mathsf{Enc}(m, \mathsf{pk})$: *A (probabilistic) algorithm that takes input a plaintext* $m \in \mathcal{M}_{\mathsf{pk}}$ *and a public key* pk*. It returns the ciphertext* $c \in \mathcal{C}_{\mathsf{pk}}$ *of* m *under* pk*.*

$\mathsf{Dec}(c, \mathsf{sk})$: *A deterministic algorithm that takes input a ciphertext* $c \in \mathcal{C}_{\mathsf{pk}}$ *and a private key* sk*. It returns the plaintext* $m \in \mathcal{M}_{\mathsf{pk}}$ *of* c *under* sk*.*

A public-key-encryption scheme needs to be *correct* and to satisfy at least *indistinguishability against chosen-plaintext attacks (IND-CPA)* in order to be *secure*.

Definition 2.18 (Correctness)**.** *A PKE scheme* PKE *is* correct *if for all security parameters* κ*, all choices of key pairs* $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa)$*, all* $m \in \mathcal{M}_{\mathsf{pk}}$ *it holds that*

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

IND-CPA security considers only passive adversaries. It is equivalent to semantic security, which demands that no PPT adversary—when given the ciphertext and the length of the unknown plaintext—is able to derive any additional information on the plaintext with non-negligible probability.

Definition 2.19 (IND-CPA)**.** *A PKE scheme* PKE *is called* IND-CPA secure*, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{ll} (\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa),\ b \xleftarrow{R} \{0,1\}, & \\ (\mathsf{st}, m_0, m_1) \xleftarrow{R} \mathcal{A}(\mathsf{pk}), & \\ c \xleftarrow{R} \mathsf{Enc}(m_b, \mathsf{pk}), & : \quad b^* = b \\ b^* \xleftarrow{R} \mathcal{A}(\mathsf{st}, c) & \end{array}\right] - \frac{1}{2} \leq \epsilon(\kappa).$$

Stronger models are indistinguishability against chosen-ciphertext attacks (IND-CCA) [NY90] and the even stronger indistinguishability against adaptive chosen-ciphertext attacks (IND-CCA2) [RS92]. The former allows access to a decryption oracle before the adversary outputs the challenge ciphertext c^*, while the latter additionally allows oracle access (for ciphertexts different from c^*) after c^* has been set.

2.5 Commitments

Commitments [Blu81] are the digital analogue to sealed envelopes: Their content stays hidden and remains unchanged, but can be opened later if required.

Definition 2.20 (Commitment scheme)**.** *A* commitment scheme CS *consists of the following PPT algorithms:*

$\mathsf{Setup}(1^\kappa)$: *A (probabilistic) algorithm that takes input a security parameter* 1^κ*. It outputs public parameters* pp *(for message space* $\mathcal{M}_{\mathsf{pp}}$*).*

$\mathsf{Commit}(\mathsf{pp}, m)$: *A (probabilistic) algorithm that takes input the public parameters* pp *and a value* $m \in \mathcal{M}_{\mathsf{pp}}$*. It outputs a tuple* (C, O) *representing the commitment* C *to* m *and opening* O*.*

$\mathsf{Open}(\mathsf{pp}, C, O)$: *A deterministic algorithm that takes input the public parameters* pp*, a commitment* C *and an opening* O*. It outputs either* $m \in \mathcal{M}_{\mathsf{pp}}$ *or* $\perp$ *to indicate success or failure, respectively.*

A commitment scheme is secure if it is *correct*, *hiding* and *binding*. Informally, hiding means that the value $m \in \mathcal{M}_{\mathsf{pp}}$ is hidden in C unless the opening O is available, whereas binding means that it is impossible to find a second opening O' such that C opens to $m' \neq m$. Moreover, a commitment scheme is a *trapdoor-commitment scheme*, if (in a loose sense) there exists a trapdoor that allows us to arbitrarily open a given commitment.

Definition 2.21 (Correctness)**.** *A commitment scheme* CS *is* correct *if for all security parameters* κ*, all choices of public parameters* $\mathsf{pp} \xleftarrow{R} \mathsf{Setup}(1^\kappa)$*, all* $m \in \mathcal{M}_{\mathsf{pp}}$ *it holds that:*

$$\Pr\left[\, \mathsf{Open}(\mathsf{pp}, \mathsf{Commit}(\mathsf{pp}, m)) = m \,\right] = 1.$$

Definition 2.22 (Binding)**.** *A commitment scheme* CS *is* binding*, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l} \mathsf{pp} \xleftarrow{R} \mathsf{Setup}(1^\kappa), \\ (C, O, O') \xleftarrow{R} \mathcal{A}(\mathsf{pp}), \\ m \leftarrow \mathsf{Open}(\mathsf{pp}, C, O), \\ m' \leftarrow \mathsf{Open}(\mathsf{pp}, C, O') \end{array} \;:\; m \neq m' \;\wedge\; m, m' \neq \perp \right] \leq \epsilon(\kappa).$$

A commitment scheme is *perfectly binding*, if the above game holds for unbounded adversaries and $\epsilon \equiv 0$.

Definition 2.23 (Hiding)**.** *A commitment scheme* CS *is* hiding*, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l} \mathsf{pp} \xleftarrow{R} \mathsf{Setup}(1^\kappa),\ b \xleftarrow{R} \{0,1\}, \\ (\mathsf{st}, m_0, m_1) \xleftarrow{R} \mathcal{A}(\mathsf{pp}), \\ (C, \cdot) \xleftarrow{R} \mathsf{Commit}(\mathsf{pp}, m_b), \\ b^* \xleftarrow{R} \mathcal{A}(\mathsf{st}, C) \end{array} \;:\; b = b^* \right] - \frac{1}{2} \leq \epsilon(\kappa).$$

A commitment scheme is *perfectly hiding*, if the above game holds for unbounded adversaries and $\epsilon \equiv 0$.

Well-known examples for commitments are DL commitments and Pedersen commitments [Ped92]. While former are perfectly binding and hiding under

the DL assumption only in a weaker model (i.e., with respect to random messages), latter are perfectly hiding and computationally binding (under the DL assumption). Moreover, any IND-CPA-secure encryption scheme yields a perfectly binding and computationally hiding commitment scheme. In Section 2.6.1, we will see how to build perfectly hiding commitments from Σ-protocols.

2.5.1 Generalized Pedersen Commitments

Now, we will discuss the generalized Pedersen commitment scheme [Ped92], which allows to commit to a vector of messages $\vec{m} = (m_i)_{i\in[n]} \in \mathbb{Z}_p^n$ and gives the Pedersen commitment scheme when instantiated with $n = 1$. Generalized Pedersen commitments are perfectly hiding, computationally binding under the DL assumption and *length-reducing*, that is, the commitment is always a single group element irrespective of n. For further use throughout this thesis, we

Scheme 1 The generalized Pedersen commitment scheme.

$\mathsf{Setup}(1^\kappa, 1^n)$: Given a security parameter 1^κ and a vector length n in unary, choose a group $\mathbb{G}$ of prime order p with $\log_2 p = \lceil\kappa\rceil$ and $n+1$ distinct generators $(P_i)_{i\in[n]}, Q \in \mathbb{G}$ and output public parameters $\mathsf{pp} \leftarrow (\mathbb{G}, p, (P_i)_{i\in[n]}, Q)$.

$\mathsf{Commit}(\mathsf{pp}, \vec{m}; r)$: Given public parameters $\mathsf{pp} = (\mathbb{G}, p, (P_i)_{i\in[n]}, Q)$, a vector $\vec{m} \in \mathbb{Z}_p^n$ and randomness $r \in \mathbb{Z}_p$, output a commitment $C \leftarrow \sum_{i\in[n]} m_i P_i + rQ$ and an opening $O \leftarrow (\vec{m}, r)$.

$\mathsf{Open}(\mathsf{pp}, C, O)$: Given public parameters $\mathsf{pp} = (\mathbb{G}, p, (P_i)_{i\in[n]}, Q)$, a commitment $C \in \mathbb{G}$ and opening $O = (\vec{m}, r)$, if $C = \sum_{i\in[n]} m_i P_i + rQ$ then output $\vec{m} = (m_i)_{i\in[n]}$ and $\perp$ otherwise.

introduce the predicate $\mathsf{Check}(\mathsf{pp})$ to check for valid parameters: For a generalized Pedersen commitment in $\mathbb{G}$, we have $\mathsf{pp} = ((P_i)_{i\in[n]}, Q)$. It returns 1 if all elements in pp are in $\mathbb{G}$ and pairwise distinct; and 0 otherwise.

2.6 Zero-Knowledge Proofs of Knowledge

In the following, we need the definition of an interactive proof system:

Definition 2.24 (Interactive proof system (IPS))**.** *An* IPS $(\mathcal{P}, \mathcal{V})$ for a language L *is an interactive protocol between an unrestricted prover* $\mathcal{P}$ *and a PPT verifier* $\mathcal{V}$ *such that the following conditions hold:*

Completeness: $\forall\, x \in L : \Pr[\,(\mathcal{P}, \mathcal{V})(x) = 1\,] = 1$.

Soundness: $\forall\, x \notin L\ \forall\, \mathcal{P}^* : \Pr[\,(\mathcal{P}^*, \mathcal{V})(x) = 1\,] \leq \frac{1}{2}$,

where $\mathcal{P}^*$ can be any (malicious) prover and $(\mathcal{P},\mathcal{V})(x) = 1$ denotes that $\mathcal{V}$ accepts the interaction with $\mathcal{P}$ on common input x.

Loosely speaking, an IPS is required to be *complete*, that is, an honest prover can always convince the verifier, and *sound*, i.e., any dishonest prover can only convince a verifier with a certain probability.

2.6.1 Zero-Knowledge Proofs

Zero-knowledge proofs (ZKPs) [GMR85] are IPSs, where a prover $\mathcal{P}$ interacts with a verifier $\mathcal{V}$ on some common input x and is able to convince the verifier of x being contained in some formal language L without $\mathcal{V}$ learning anything beyond the validity of the proven statement, i.e., the language membership. This is formalized as follows:

Definition 2.25 (Zero knowledge (ZK))**.** *An* IPS $(\mathcal{P},\mathcal{V})$ for a language L *is* ZK *if for any (malicious) verifier $\mathcal{V}^*$, there exists a PPT algorithm $\mathcal{S}$ (the* simulator*) such that:*

$$\{\mathcal{S}(x)\}_{x\in L} \approx \{\langle(\mathcal{P},\mathcal{V}^*)(x)\rangle\}_{x\in L},$$

where $\langle(\mathcal{P},\mathcal{V}^*)(x)\rangle$ denotes the transcript of the interaction between $\mathcal{P}$ and $\mathcal{V}^*$ on common input x.

In fact, statements from arbitrary NP-languages $L \in \mathrm{NP}$ can efficiently be proven in ZK, as there are known ZKPs for NP-complete languages such as 3-coloring [GMW87].

2.6.2 Proofs of Knowledge

In cryptography, we are often interested in IPSs having a stronger soundness definition denoted as *proofs of knowledge (PoKs)* [BG93]. For our discussion let $L_{\mathcal{R}} = \{x : \exists w\colon (x,w) \in \mathcal{R}\} \subseteq \{0,1\}^*$ be a formal language, where $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ is a binary, polynomial-time (witness) relation. For such a relation, the membership of $x \in L_{\mathcal{R}}$ can be decided in polynomial time (in $|x|$), when given a witness w certifying $(x,w) \in \mathcal{R}$, which is of polynomial length in $|x|$. Contrary, to proof systems as defined above, PoKs formalize the notion of knowledge: Instead of just proving the validity of a statement, they prove the knowledge of a witness w certifying the validity. In doing so, PoKs give a strong and straightforward way to define soundness: If a machine $\mathcal{M}$ "knows something" and proves this knowledge to someone else, then we can as well extract this witness from $\mathcal{M}$. This is formalized by another machine $\mathcal{E}$, the extractor, which when given access to $\mathcal{M}$ is able to extract a witness from $\mathcal{M}$.

Definition 2.26 (Proof of knowledge (PoK))**.** *Let $k\colon \{0,1\}^* \to [0,1]$ be a function and $\mathcal{R}$ be a binary, polynomial-time witness relation. A* PoK $(\mathcal{P},\mathcal{V})$ for relation $\mathcal{R}$ with knowledge error k *is an IPS such that besides completeness the following condition holds:*

Knowledge soundness: *There exists a $c > 0$ and an expected PPT algorithm $\mathcal{E}$ (the* extractor*) having oracle access to any (malicious) prover $\mathcal{P}^*$ such that for every $x \in L_{\mathcal{R}}$ the following holds. Let $\delta(x)$ be the probability that $\mathcal{V}$ accepts input x after interacting with $\mathcal{P}^*$. If $\delta(x) > k(x)$, then on input x and oracle access to $\mathcal{P}^*$, the machine $\mathcal{E}$ outputs a string w such that $(x, w) \in \mathcal{R}$ within an expected number of steps bounded by*

$$\frac{|x|^c}{\delta(x) - k(x)}.$$

Let us consider an example for a group-theoretic language, which often appears in cryptographic applications:

Example 2.27 (DL proof)**.** A statement proven by a PoK could, for instance, be the knowledge of a private key $w \in \mathbb{Z}_p$ corresponding to some public key $Y = wP$ in a group $\mathbb{G} = \langle P \rangle$ of prime order p (here the relation would be $\mathcal{R}_{DL} = \{(Y, w) : w \in \mathbb{Z}_p, Y = wP \in \mathbb{G}\}$).

We use the common notation of [CS97] and give a short example to illustrate it:

Example 2.28 (Notation)**.** Let the setting be as in Example 2.27. We denote a PoK of a discrete logarithm $w = \log_P Y$ as $\mathsf{PoK}\{\alpha : Y = \alpha P\}$, where Greek letters stand for witnesses and all other involved values are public.

A formulation of the ZK property, which considers a simulator working for any (malicious) verifier $\mathcal{V}^*$ and which we are going to use subsequently, is the following one:

Definition 2.29 (Universally simulatable ZK)**.** *A PoK $(\mathcal{P}, \mathcal{V})$ for a relation $\mathcal{R}$ is* (universally simulatable) ZK *if there exists an (expected) PPT algorithm $\mathcal{S}$ (the* simulator*) such that when having oracle access to any (malicious) PPT verifier $\mathcal{V}^*$ it holds that:*

$$\{\mathcal{S}^{\mathcal{V}^*}(x)\}_{x \in L_{\mathcal{R}}} \approx \{\langle(\mathcal{P}(x, w), \mathcal{V}^*(x))\rangle\}_{(x,w) \in \mathcal{R}}.$$

We call a PoK $(\mathcal{P}, \mathcal{V})$ *zero-knowledge proof of knowledge (ZKPoK)*, if it fulfills Definition 2.29.

An important property that is weaker than ZK is *witness indistinguishability (WI)*. Informally, it says that one cannot tell which witness was used during the conduction of a proof.

Definition 2.30 (Witness indistinguishability (WI))**.** *A PoK $(\mathcal{P}, \mathcal{V})$ for a relation $\mathcal{R}$ is called* witness indistinguishable (WI)*, if for all (x, w, w') with $(x, w) \in \mathcal{R}$ and $(x, w') \in \mathcal{R}$ and for any (malicious) PPT verifier $\mathcal{V}^*$ it holds that:*

$$\langle(\mathcal{P}(x, w), \mathcal{V}^*(x))\rangle \approx \langle(\mathcal{P}(x, w'), \mathcal{V}^*(x))\rangle.$$

We call a PoK $(\mathcal{P}, \mathcal{V})$ *witness-indistinguishable proof of knowledge (WIPoK)*, if it fulfills Definition 2.30.

2.6.3 Σ-Protocols and ZKPoKs from Σ-Protocols

Σ-protocols are efficient instantiations of PoKs [Sch90]. A Σ-protocol $\Pi = (\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2), \mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2))$ is a *3-move public-coin honest-verifier* zero-knowledge proof of knowledge (ZKPoK).

These three moves can be outlined as follows. The first message sent by prover $\mathcal{P}$ is $C \xleftarrow{R} \mathcal{P}_1(x, w)$, on which $\mathcal{V}_1(x, C)$ responds with a random challenge $c \xleftarrow{R} \{0,1\}^\kappa$ (with κ being a security parameter).[5] $\mathcal{P}$'s second message is $s \xleftarrow{R} \mathcal{P}_2(x, w, C, c)$. In the end, $\mathcal{V}_2(x, C, c, s)$ outputs 1 or 0 indicating whether it accepted the proof or not. Public coin means that the prover has access to the verifier's random coins (used for generating the challenge).

A Σ-protocol is *secure* if it is *complete*, *special-sound* and *special-honest-verifier ZK*. *Completeness* is straightforward and we now give the intuition behind the remaining properties. *Special soundness* states that there exists a PPT extraction algorithm $\mathcal{E}$, which when given transcripts of two accepting runs for $x \in L_\mathcal{R}$ having identical first messages—say, $\tau = (C, c, s)$ and $\tau' = (C, c', s')$ with $c \neq c'$—can recover a witness w such that $(x, w) \in \mathcal{R}$. *Special-honest-verifier ZK* means that the ZK property only holds in front of an honest verifier, which chooses its challenge uniformly random. It requires that there is a PPT simulation algorithm $\mathcal{S}$ that given x and c produces transcripts (C, c, s) satisfying the same probability distribution as transcripts originating from protocol runs between $\mathcal{P}$ and $\mathcal{V}$ when run on common input x.

More formally, a Σ-protocol is defined as follows:

Definition 2.31 (Σ-Protocol)**.** *An IPS $\Pi = (\mathcal{P}, \mathcal{V})$ is a* Σ*-protocol for a relation $\mathcal{R}$ with challenge length $\kappa > 0$, if it is a 3-move public-coin protocol and the following requirements hold:*

Completeness: $\forall\,(x, w) \in \mathcal{R} : \Pr[\,(\mathcal{P}(x, w), \mathcal{V}(x)) = 1\,] = 1$.

Special soundness: *There exists a PPT extractor $\mathcal{E}$ that on input x and any pair $\tau = (C, c, s)$, $\tau' = (C, c', s')$ of accepting transcripts for protocol runs on x with distinct $c, c' \in \{0,1\}^\kappa$ outputs w such that $(x, w) \in \mathcal{R}$.*

Special honest-verifier ZK: *There exists a PPT simulator $\mathcal{S}$ such that the following holds:*

$$\{\mathcal{S}(x, c)\}_{(x,c)\in L_\mathcal{R}\times\{0,1\}^\kappa} \approx \{\langle(\mathcal{P}(x, w), \mathcal{V}(x; c))\rangle\}_{(x,w,c)\in\mathcal{R}\times\{0,1\}^\kappa}.$$

The following statements characterize Σ-protocols: Lemma 2.32 states that Σ-protocols achieve a negligible soundness error and Lemma 2.33 says that every Σ-protocol is a WIPoK.

Lemma 2.32. *Let Π be a Σ-protocol for a relation $\mathcal{R}$ with challenge length κ. Then, Π is a PoK with knowledge error $2^{-\kappa}$.*

[5] In our definitions, we will assume all challenges to be bitstrings; whereas in our examples (and in context of our schemes later on) challenges will, in fact, be values from $\mathbb{Z}_p$.

Lemma 2.33. *Every Σ-protocol is perfectly WI.*

Let us now take a closer look at the Σ-protocol for $\mathsf{PoK}\{\alpha : Y = \alpha P\}$:

Example 2.34 (Schnorr protocol [Sch90])**.** Again, let the setting be as in Example 2.27. Here, $\mathcal{P}$ starts by choosing $k \xleftarrow{R} \mathbb{Z}_p$ and sending a DL commitment $C \leftarrow kP$ to $\mathcal{V}$. $\mathcal{V}$ responds with a challenge $c \xleftarrow{R} \mathbb{Z}_p$, on which $\mathcal{P}$ sends $s \leftarrow k + wc \mod p$. In the end, $\mathcal{V}$ verifies whether $sP = C + cY$.

The transcript of this interaction is $\tau = (C, c, s)$. An extractor $\mathcal{E}$ is allowed to rewind $\mathcal{P}$ to the step after sending C and can so derive a second transcript $\tau' = (C, c', s')$ by sending a different challenge c'. From these two transcripts, $\mathcal{E}$ can now extract witness $w = \frac{s'-s}{c'-c}$.

With regard to the honest-verifier zero-knowledge (HVZK) property, a simulator $\mathcal{S}$ on input $(Y, c) \in \mathbb{G} \times \mathbb{Z}_p$ can always create simulated transcripts $\tau = (C, c, s)$ having the same distribution as transcripts originating from a real interaction by picking $s \xleftarrow{R} \mathbb{Z}_p$ and computing $C \leftarrow sP - cY$.

Σ-protocols can be composed [CDS94], yielding the ability to conduct proofs over more complex statements: e.g., conjunctions, disjunctions of proofs; range proofs, equality proofs, etc. In fact, the composition of Σ-protocols is again a Σ-protocol. Subsequently, we will sketch the AND- and the OR-composition of Σ-protocols, which allow us to conduct proofs using conjunctive and disjunctive relations, respectively. For further details, we refer the reader to [CDS94, Sch15].

When applying the Fiat-Shamir transform [FS87] to Σ-protocols, we can obtain non-interactive zero-knowledge proofs (NIZKPs) in the ROM. Here, challenge c is essentially replaced by a hash value of the transcript so far.

AND-Composition. We will now briefly discuss the AND-composition of Σ-protocols. Let $\mathcal{R}'$ and $\hat{\mathcal{R}}$ be two relations having Σ-protocols with challenge length $\kappa > 0$; say Π and $\hat{\Pi}$, respectively. A Σ-protocol Π for a relation $\mathcal{R} = \{((x', \hat{x}), (w', \hat{w})) : (x', w') \in \mathcal{R}' \wedge (\hat{x}, \hat{w}) \in \hat{\mathcal{R}}\}$ (with challenge length κ) can be built efficiently by running Π' and $\hat{\Pi}$ in parallel using a common challenge $c \in \{0,1\}^\kappa$.

OR-Composition. We will now outline the OR-composition of Σ-protocols, as it is of particular interest in the following.

Let $\mathcal{R}'$ and $\hat{\mathcal{R}}$ be two relations having Σ-protocols with challenge length $\kappa > 0$; say $\Pi' = (\mathcal{P}' = (\mathcal{P}'_1, \mathcal{P}'_2), \mathcal{V}' = (\mathcal{V}'_1, \mathcal{V}'_2))$ and $\hat{\Pi}$, respectively. A Σ-protocol $\Pi = (\mathcal{P}, \mathcal{V})$ for a relation $\mathcal{R} = \{((x', \hat{x}), (w', \hat{w})) : (x', w') \in \mathcal{R}' \vee (\hat{x}, \hat{w}) \in \hat{\mathcal{R}}\}$ (of challenge length κ) can be built efficiently by composing a simulated proof for one clause in the disjunction with a regular proof for the other clause.

More precisely, suppose that $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2)$ and $\mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2)$ run on common input $x = (x', \hat{x}) \in L_{\mathcal{R}}$ and $\mathcal{P}$ additionally on secret input w' such that $(x', w') \in \mathcal{R}'$. Then, $\mathcal{P}_1$ on input (x, w') runs $C' \xleftarrow{R} \mathcal{P}'_1(x', w')$ and simulator $(\hat{C}, \hat{c}, \hat{s}) \xleftarrow{R} \hat{\mathcal{S}}(\hat{x}, \hat{c})$ of $\hat{\Pi}$ to obtain a simulated transcript by picking $\hat{c} \xleftarrow{R} \{0,1\}^\kappa$. $\mathcal{P}_1$ sends first message $(C', \hat{C})$ to $\mathcal{V}_1$ and $\mathcal{P}_2$ gets in return challenge $c \in \{0,1\}^\kappa$.

Next, $\mathcal{P}_2$ sets $c' \leftarrow c \oplus \hat{c}$, runs $s' \xleftarrow{R} \mathcal{P}'_2(x', w', C', c')$ and sends final message $(s', c', \hat{s}, \hat{c})$ to $\mathcal{V}_2$. If $c \neq c' \oplus \hat{c}$, then $\mathcal{V}_2$ stops and returns 0. Else, $\mathcal{V}_2$ runs $\mathcal{V}'_2(x', C', c', s')$ and $\hat{\mathcal{V}}_2(\hat{x}, \hat{C}, \hat{c}, \hat{s})$ and outputs 1 or 0 indicating whether both $\mathcal{V}'_2$ and $\hat{\mathcal{V}}_2$ accepted the proof or not.

WI ensures that one cannot distinguish whether a witness satisfying the first or the second clause has been used.

Commitments from Σ-Protocols

We will now sketch the observation from [FS90, Dam90] that any Σ-protocol $\Pi = (\mathcal{P}, \mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2))$ for a relation $\mathcal{R}$ implies a commitment scheme. Let $\kappa > 0$ be the challenge length of Π. The setup algorithm picks $(x, w) \xleftarrow{R} \mathcal{R}$ and sets the commitment parameters $\mathsf{pp} \leftarrow x$ (witness w is the trapdoor). In order to commit to a value $c \in \{0,1\}^\kappa$, the Commit algorithm runs the special simulator $\mathcal{S}$ of Π on input (x, c), which produces a transcript (C, c, s), and outputs C as commitment to c and $O = (c, s)$ as its opening. The Open algorithm returns opening c if $\mathcal{V}_2(x, C, c, s) = 1$ and $\perp$ otherwise.

The scheme is perfectly hiding and computationally binding (as long as w remains secret).

Such a commitment itself defines a relation $\mathcal{R}' = \{((x, C), (c, s)) : x \in L_\mathcal{R}, \mathcal{V}_2(x, C, c, s) = 1\}$, which is comprised of tuples containing a commitment and its opening. This is also known as the *commitment relation*.

ZKPoKs from Σ-Protocols in the Standard Model

Subsequently, we are going to sketch the ZKPoKs from [CDM00]. These are very efficient 4-move perfect ZKPoKs for a relation $\mathcal{R}$ from Σ-protocols and their commitments (satisfying Definition 2.29 with rewindable black-box access to verifier $\mathcal{V}^*$).

More precisely, they consider the case where both relation $\mathcal{R}$ and the related commitment relation $\mathcal{R}'$ have Σ-protocols with challenge length κ. By nesting two Σ-protocols, they are then able to construct 4-move perfect ZKPoKs for relation $\mathcal{R}$ with knowledge error $2^{-\kappa}$.

This allows, in particular, ZKPoKs over DLs (and other relations built from q-one-way functions [CD98]).

Outline. For the sake of an easy presentation, we discuss the 6-move variant (as also done in [CDM00]; the 4-move protocol can be obtained by collapsing moves 2 and 3 of Part 1 and moves 1 and 2 of Part 2).

In the following, a prover $\mathcal{P}$ tries to prove knowledge of a witness w to some $x \in L_\mathcal{R}$.

Part 1: On input x, verifier $\mathcal{V}$ creates a commitment C (for public parameters $\mathsf{pp} \leftarrow x$) to a value $e \in \{0,1\}^\kappa$ and proves knowledge of C's opening O to prover $\mathcal{P}$ by using a Σ-protocol $\Pi_\mathcal{V}$ for the corresponding commitment relation $\mathcal{R}'$. If $\Pi_\mathcal{V}$ fails, then $\mathcal{P}$ aborts. ($\Pi_\mathcal{V}$ does not give any information about e, as $\Pi_\mathcal{V}$

is WI, when assuming that there is an overwhelming number of openings—the opening to e is just one of them.)

Part 2: $\mathcal{P}$ considers the relation $\mathcal{R}_\vee = \{((x,x'),(w,w')) : (x,w) \in \mathcal{R} \vee (x',w') \in \mathcal{R}'\}$: On input (x,w), $\mathcal{P}$ proves either knowledge of witness w for statement x or knowledge of an opening of commitment C. In other words, $\Pi_\mathcal{P}$ is the OR-composition of a Σ-protocol Π for relation $\mathcal{R}$ and a Σ-protocol Π' for the related commitment relation $\mathcal{R}'$. (Since $\mathcal{P}$ does not know how to open commitment C, $\mathcal{P}$ has to simulate Π' and perform Π honestly.)

Properties. We will now give the intuition behind the knowledge soundness and the ZK property of the described protocol.

The knowledge soundness of the described protocol is unconditional; this follows from the following knowledge extractor $\mathcal{E}$. $\mathcal{E}$ takes the role of an honest verifier; initially generates a commitment C with opening O and runs the protocol with prover $\mathcal{P}^*$ by conducting $\Pi_\mathcal{V}$. Then, by rewinding $\mathcal{P}^*$ during the conduction of $\Pi_\mathcal{P}$, $\mathcal{E}$ extracts either a witness w for x and we are done; or an opening O' of commitment C. When assuming that there is an overwhelming number of different openings of C, then by the WI of $\Pi_\mathcal{V}$, O' is different from O with overwhelming probability. Using two different openings O, O' of C, $\mathcal{E}$ is now able to extract a witness for x and we are done as well.

The protocol satisfies the ZK property (Definition 2.29): Intuitively, simulator $\mathcal{S}$ first rewinds the verifier $\mathcal{V}^*$ in order to extract an opening O to commitment C (and, thus, a witness $w' = O$ such that $(x',w') = ((x,C),O) \in \mathcal{R}'$) and then conducts $\Pi_\mathcal{P}$ for the second part of the disjunction, that is, $\mathcal{S}$ proves knowledge of a witness to commitment relation $\mathcal{R}'$. By the perfect WI of $\Pi_\mathcal{P}$, $\mathcal{V}^*$ cannot distinguish whether $\mathcal{S}$ has conducted the proof with regard to relation $\mathcal{R}$ or with regard to commitment relation $\mathcal{R}'$. In total, this yields perfect ZK.

Concurrent ZKPoKs from Σ-Protocols in the CRS Model

Concurrent ZK guarantees, loosely speaking, that a protocol's ZK property holds even in front of arbitrarily interleaved parallel protocol runs.

We will now briefly mention the result from [Dam00], which proposes a generic transform from Σ-protocols to concurrent ZKPoKs: Under the assumption of OWFs and at the expense of a CRS, it shows how to convert any Σ-protocol for an arbitrary NP-relation $\mathcal{R}$ into a 3-move concurrent ZKPoK for relation $\mathcal{R}$ (without imposing any timing constraints). The protocol employs a trapdoor commitment and, essentially, the simulation is based on the trapdoor and hiding properties of used commitment scheme. Furthermore, the protocol is concurrent ZK, as the simulator does not require rewinding the verifier $\mathcal{V}^*$.

2.7 Digital Signatures

Digital signatures (DSs) are cryptographic primitives providing *non-repudiation*, *integrity*, *authenticity* of messages exchanged over an insecure channel. Thereby, the sender (or signer) holds a secret key sk that she uses to sign messages and a corresponding public key pk that uniquely identifies her (when assuming the availability of a public-key infrastructure (PKI)). Then, integrity guarantees that when receiving a signed message it can be detected whether the message has not been altered in transit; authenticity gives the receiver (or verifier) reason to believe that the message stems from the alleged sender and non-repudiation guarantees that the sender cannot deny having signed a message.

The formal definition of a digital signature scheme is as follows:

Definition 2.35 (Digital signature (DS) scheme)**.** *A* DS scheme DS *is a tuple of the following PPT algorithms:*

$\mathsf{KeyGen}(1^\kappa)$: *A probabilistic algorithm that takes input a security parameter* 1^κ *and outputs a private key* sk *and a public key* pk *(we assume that* pk *includes a description of the message space* $\mathcal{M}_{\mathsf{pk}}$*).*

$\mathsf{Sign}(m, \mathsf{sk})$: *A (probabilistic) algorithm that takes input a message* $m \in \mathcal{M}_{\mathsf{pk}}$*, a secret key* sk *and outputs a signature* σ *under* sk *on* m*.*

$\mathsf{Verify}(m, \sigma, \mathsf{pk})$: *A deterministic algorithm that takes input a message* $m \in \mathcal{M}_{\mathsf{pk}}$*, a signature* σ*, a public key* pk *and outputs* 1 *if* σ *is a valid signature for* m *under* pk *and* 0 *otherwise.*

A DS scheme is secure, if it is *correct* and *existentially unforgeable under adaptive chosen-message attacks (EUF-CMA)* [GMR88]. Correctness ensures that every honestly computed signature under honestly generated keys always verifies and unforgeability implies the properties integrity, authenticity and non-repudiation. We define these properties below.

Definition 2.36 (Correctness)**.** *A DS scheme* DS *is* correct*, if for all security parameters* κ*, all choices of key pairs* $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa)$*, all* $m \in \mathcal{M}_{\mathsf{pk}}$ *we have:*

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

Definition 2.37 (EUF-CMA)**.** *A DS scheme* DS *is* EUF-CMA secure*, if for all PPT adversaries* $\mathcal{A}$ *having access to a signing oracle* $\mathsf{Sign}(\cdot, \mathsf{sk})$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l|r} (\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa), & \mathsf{Verify}(m^*, \sigma^*, \mathsf{pk}) = 1 \\ (m^*, \sigma^*) \xleftarrow{R} \mathcal{A}^{\mathsf{Sign}(\cdot, \mathsf{sk})}(\mathsf{pk}) & \wedge\ m^* \notin Q \end{array}\right] \leq \epsilon(\kappa),$$

where Q *is the set of queries which* $\mathcal{A}$ *has issued to the signing oracle.*

This definition is the standard notion of unforgeability. An even stronger notion is called *strongly existentially unforgeable under adaptive chosen-message attacks (sEUF-CMA)*. It requires $\mathcal{A}$ to output a message-signature pair (m^*, σ^*) that is different from all queried message-signature pairs. Other, weaker notions consider *random-message attacks* ($\mathcal{A}$ can only query signatures on random messages that are not under the control of the adversary), *known-message attacks* ($\mathcal{A}$ is only given signatures on a set of messages that it can define a priori) or *key-only attacks* ($\mathcal{A}$ can only access the public key) [Kat10].

Remark 2.38. Note that Σ-protocols (cf. Section 2.6) and DS schemes are closely related [CS97, CL06]. Applying the Fiat-Shamir transform [FS87] to a Σ-protocol, gives a straightforward way to build DS schemes that are EUF-CMA secure in the ROM. So, for instance, the Schnorr signature scheme [Sch90] can be derived from the Schnorr Σ-protocol using the Fiat-Shamir transform.

2.7.1 Structure-Preserving Signatures

Structure-preserving signatures (SPSs) [AFG+10] have originally been introduced in the context of Groth-Sahai (GS) proofs [GS08]; as new type of signature schemes compatible with this proof system defined on bilinear groups. Later on, other applications of structure-preserving signature (SPS), which are of independent interest, have been found as well [LPJY13, HS14, FHS15a].

An SPS is a signature scheme defined in the context of a bilinear group $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$. Such a scheme allows to sign group-element vectors without requiring any a-priori encoding. The public keys and the signatures σ themselves are group elements and verification is done solely by means of group-membership tests and the conjunction of pairing-product equations (PPEs). In case of fully structure-preserving signatures even secret keys are group elements [AKOT15].

The abstract model of a Type-3 SPS scheme, i.e., an SPS defined for Type-3 bilinear groups, is as follows.

Definition 2.39 (Structure-preserving signature (SPS) scheme)**.** *An* SPS scheme SPS *consists of the following PPT algorithms:*

$\mathsf{Setup}(1^\kappa)$: *A (probabilistic) setup algorithm that takes input a security parameter* 1^κ*. It outputs public parameters* pp *containing a bilinear group* $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$ *of prime order* p *with* $\log_2 p = \lceil \kappa \rceil$*, elements of* $\mathbb{Z}_p$ *and group elements of* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*.*

$\mathsf{KeyGen}(\mathsf{pp})$: *A probabilistic algorithm that takes input public parameters* pp*. It outputs a key pair* $(\mathsf{sk}, \mathsf{pk})$*, where* pk *contains* pp *and group elements from* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*.*

$\mathsf{Sign}(M, \mathsf{sk})$: *A (probabilistic) algorithm that takes input a message* M *consisting of elements from* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *and a secret key* sk*. It outputs a signature* σ *for the message* M*, which is comprised of elements from* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*.*

$\mathsf{Verify}(M, \sigma, \mathsf{pk})$: *A deterministic algorithm that takes input a message-signature pair* (M, σ) *and a public key* pk. *It checks the validity of* (M, σ) *using a conjunction of group-membership tests and PPEs and outputs* 1 *if* σ *is valid for* M *under* pk *and* 0 *otherwise.*

Similarly, we can define Type-1 and Type-2 SPSs, i.e., SPSs over Type-1 and Type-2 bilinear groups, respectively. For a Type-2 SPS, the verification is additionally allowed to evaluate the homomorphism $\Psi\colon \mathbb{G}_2 \to \mathbb{G}_1$.

An SPS scheme SPS is secure, if it is *correct* and *EUF-CMA secure*. The definitions are analogous to those in Section 2.7, where the correctness definition additionally has to take account of the Setup algorithm and the different input behavior of KeyGen.

In [AGHO11], Abe et al. showed that Type-3 SPSs having constant-size signatures cannot exist, unless the signature has at least 3 elements, its elements stem from both groups *(bilateral)* and the SPS scheme uses at least 2 PPEs for verification; in [AGOT14a], they showed similar results for Type-2 SPSs. Moreover, in [AGO11], Abe et al. proved that the unforgeability of optimally short Type-3 SPS schemes (i.e., with 3-element signatures) cannot be reduced to non-interactive assumptions. This means that the unforgeability of such schemes can only be proven in the GGM. In this context, Abe et al. [AGHO11] also introduced the notion of *generic signers*, i.e., signing algorithms that create signatures solely via a sequence of generic-group operations. This seems to be a natural view on SPS signing algorithms and applies to all SPSs known so far [Fuc09, AHO10, AFG+10, AGHO11, CDH12, AGOT14b, LPY15, KPW15, Gha16].

3

Structure-Preserving Signatures on Equivalence Classes

The obvious mathematical breakthrough would be development of an easy way to factor large prime numbers.
— Bill Gates

This chapter introduces structure-preserving signature on equivalence classes (SPS-EQ) and their formal models alongside with two constructions; one GGM and one standard-model construction. Furthermore, it details relations to SPS and gives an impossibility result for SPS-EQ; separating certain SPS-EQ variants from non-interactive assumptions.

This chapter relies on joint work with Georg Fuchsbauer and Daniel Slamanig. The following material stems (sometimes verbatim) from [HS14, FHS14, FHS15a, FHS15b, FHS16]. Note that the black-box separation in Section 3.6 is unpublished work.

3.1 Basic Idea

Structure-preserving signatures on equivalence classes (SPS-EQs) are a new type of SPSs. Contrary to conventional SPSs, their message space is required to be unilateral (to achieve indistinguishability on the message space), that is, the direct sum $\mathbb{G}^\ell$ of several copies of a group $\mathbb{G}$. The clue is that if $\mathbb{G}$ is of prime order p, then $\mathbb{G}^\ell$ contains an underlying vector space $\mathbb{Z}_p^\ell$, which we can partition into projective equivalence classes for $\ell > 1$ by using a projective equivalence

relation. Defined on $\mathbb{G}^\ell$, this equivalence relation $\sim_{\mathcal{R}}$ is as follows: [1]

$$M \in \mathbb{G}^\ell \sim_{\mathcal{R}} N \in \mathbb{G}^\ell \iff \exists\, \mu \in \mathbb{Z}_p^* : M = \mu N$$

To give some intuition: Such an equivalence class contains all elements on a certain line running through the origin except for the all-zero vector itself.

For SPS-EQs we want a controlled form of malleability: We want that a signature on an arbitrary representative M of some class $[M]_{\mathcal{R}}$ can later be publicly updated to another representative of the same class. The main benefit of SPS-EQ is that we automatically obtain a form of indistinguishability on the message space $\mathbb{G}^\ell$ if the DDH assumption holds on $\mathbb{G}$: We cannot efficiently decide whether some message vector is random or an element of a certain equivalence class. If we additionally guarantee that updated signatures are distributed like fresh signatures (or even in a stronger sense that updated signatures are uniform in the space of signatures on the respective updated representative), then message-signature pairs falling into the same equivalence class are unlinkable.

Nevertheless, observe that direct access to the DLs in a message vector allows us to efficiently decide class membership. Thus, SPS-EQs are mainly of interest, if we either want to achieve indistinguishability in front of signers; for privacy-enhancing protocols, where the message vector is not (fully) determined by the signer (as it is the case for, e.g., blind signatures and attribute-based credentials (ABCs); cf. Chapters 4 and 7); or in protocols, where consistent and authentic randomization of values is required but indistinguishability is not important (as it is the case for, e.g., verifiably-encrypted signatures (VESs); cf. Chapter 5).

Remark 3.1. It seems possible to define other types of SPS-EQ for more general equivalence relations. For instance, $\mathbb{Z}_p^\ell$ with $\ell > 2$ can be as well factorized into equivalence classes with planes through the origin. The DDH assumption could then simply be replaced by the DLIN assumption in order to obtain indistinguishability on the message space.

3.2 Formal Definitions

We state the syntax and the security properties of an SPS-EQ. The following definition is tailored to Type-3 schemes.

Definition 3.2 (Structure-preserving signature on equivalence classes (SPS-EQ) scheme)**.** *An* SPS-EQ scheme $\mathsf{SPS\text{-}EQ}$ *over* $\mathbb{G}_i$ *consists of the following PPT algorithms:*

$\mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$: *A (probabilistic) bilinear-group generation algorithm that takes input a security parameter* 1^κ*. It outputs a Type-3 bilinear group* $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$.

[1] Actually, we restrict the message space to $(\mathbb{G}^*)^\ell$, since including the neutral element is not meaningful for our purposes

$\mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^{\ell})$: *A probabilistic algorithm that takes input a bilinear group* BG *and a vector length* $\ell > 1$ *(in unary). It outputs a key pair* $(\mathsf{sk}, \mathsf{pk})$*, where* pk *consists of elements of* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*.*

$\mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk})$: *A probabilistic algorithm that takes input a message* $M \in (\mathbb{G}_i^*)^{\ell}$ *defining an equivalence class* $[M]_{\mathcal{R}}$ *and a secret key* sk*. It outputs a signature* σ *for the representative* M *of equivalence class* $[M]_{\mathcal{R}}$*, which consists of elements of* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*.*

$\mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \mu, \mathsf{pk})$: *A probabilistic algorithm that takes input a message* $M \in (\mathbb{G}_i^*)^{\ell}$ *defining an equivalence class* $[M]_{\mathcal{R}}$*, a signature* σ*, a scalar* μ *and a public key* pk*. If* σ *is valid on* M *under* pk*, it returns an updated message-signature pair* (M', σ')*, where* $M' = \mu M \in (\mathbb{G}_i^*)^{\ell}$ *is the new representative and* σ' *its updated signature. Else, it returns* $\perp$*.*

$\mathsf{Verify}_{\mathcal{R}}(M, \sigma, \mathsf{pk})$: *A deterministic algorithm that takes input a message* $M \in (\mathbb{G}_i^*)^{\ell}$*, a signature* σ *and a public key* pk*. It performs verification using group membership tests and a conjunction of PPEs and outputs* 1 *or* 0 *indicating whether or not* σ *is valid for* M *under* pk*.*

$\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk})$: *A deterministic algorithm that takes input a secret key* sk *and a public key* pk*. It checks both keys for consistency and returns* 1 *on success and* 0 *otherwise.*

For security, we require the following properties:

Definition 3.3 (Correctness)**.** *An SPS-EQ scheme* $\mathsf{SPS\text{-}EQ}$ *over* $\mathbb{G}_i$ *is* correct *if for all* $\kappa > 0$*, all* $\ell > 1$*, all choices of bilinear groups* $\mathsf{BG} \xleftarrow{R} \mathsf{BGGen}_{\mathcal{R}}(1^{\kappa})$*, all choices of key pairs* $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^{\ell})$*, all messages* $M \in (\mathbb{G}_i^*)^{\ell}$ *and all* $\mu \in \mathbb{Z}_p^*$ *we have:*

$$\begin{aligned}
&\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk}) = 1 \quad \textit{and} \\
&\Pr\left[\, \mathsf{Verify}_{\mathcal{R}}(M, \mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk}), \mathsf{pk}) = 1 \,\right] = 1 \quad \textit{and} \\
&\Pr\left[\, \mathsf{Verify}_{\mathcal{R}}(\mathsf{ChgRep}_{\mathcal{R}}(M, \mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk}), \mu, \mathsf{pk}), \mathsf{pk}) = 1 \,\right] = 1.
\end{aligned}$$

Definition 3.4 (EUF-CMA)**.** *An SPS-EQ scheme* $\mathsf{SPS\text{-}EQ}$ *is* EUF-CMA secure*, if for all* $\ell > 1$ *and all PPT algorithms* $\mathcal{A}$ *having access to a signing oracle* $\mathsf{Sign}_{\mathcal{R}}(\cdot, \mathsf{sk})$*, there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l} \mathsf{BG} \xleftarrow{R} \mathsf{BGGen}_{\mathcal{R}}(1^{\kappa}), \\ (\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^{\ell}), \\ (M^*, \sigma^*) \xleftarrow{R} \mathcal{A}^{\mathsf{Sign}_{\mathcal{R}}(\cdot, \mathsf{sk})}(\mathsf{pk}) \end{array} : \begin{array}{c} [M^*]_{\mathcal{R}} \neq [M]_{\mathcal{R}} \quad \forall M \in Q \\ \wedge\ \mathsf{Verify}_{\mathcal{R}}(M^*, \sigma^*, \mathsf{pk}) = 1 \end{array}\right] \leq \epsilon(\kappa),$$

where Q *is the set of queries that* $\mathcal{A}$ *has issued to the signing oracle.*

We consider the following property to be a minimum requirement for SPS-EQ. Loosely speaking, it demands that after being given a random message-signature pair (M, σ), one cannot tell apart an update of (M, σ) to another arbitrary representative from a fresh random message-signature pair.

Definition 3.5 (Class-hiding)**.** *An SPS-EQ scheme* SPS-EQ *over* $\mathbb{G}_i$ *is called* class-hiding *if for all* $\ell > 1$ *and all PPT adversaries* $\mathcal{A}$ *having oracle access to* $\mathcal{O} := \{\mathcal{O}^{RM}, \mathcal{O}^{RoR}(\cdot, \mathsf{sk}, \mathsf{pk}, b)\}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l|r} \mathsf{BG} \xleftarrow{R} \mathsf{BGGen}_{\mathcal{R}}(1^\kappa),\ b \xleftarrow{R} \{0,1\}, & \\ (\mathsf{st}, \mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathcal{A}(\mathsf{BG}, 1^\ell), & b^* = b \\ b^* \xleftarrow{R} \mathcal{A}^{\mathcal{O}}(\mathsf{st}, \mathsf{sk}, \mathsf{pk}) & \wedge\ \mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk}) = 1 \end{array}\right] - \frac{1}{2} \leq \epsilon(\kappa),$$

where the oracles are defined as follows:

$\mathcal{O}^{RM}$: *Pick a message* $M \xleftarrow{R} (\mathbb{G}_i^*)^\ell$*, append it to* Q *and return* M*.*

$\mathcal{O}^{RoR}(M, \mathsf{sk}, \mathsf{pk}, b)$: *Given message* M*, key pair* $(\mathsf{sk}, \mathsf{pk})$ *and bit* b*, return* $\perp$ *if* $M \notin Q$*. On the first valid call, record* M *and* $\sigma \xleftarrow{R} \mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk})$ *and return* (M, σ)*. If later called on* $M' \neq M$*, return* $\perp$*; else pick* $R \xleftarrow{R} (\mathbb{G}_i^*)^\ell$ *and* $\mu \xleftarrow{R} \mathbb{Z}_p^*$*, set* $(M_0, \sigma_0) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \mu, \mathsf{pk})$ *and* $(M_1, \sigma_1) \xleftarrow{R} (R, \mathsf{Sign}_{\mathcal{R}}($ $R, \mathsf{sk}))$ *and return* (M_b, σ_b)*.*

In the following, we will mostly supersede this definition with two separate properties, which together imply Definition 3.5 and turn out to be more handy; a class-hiding property defined solely on the message space and requirements on the output distributions of $\mathsf{ChgRep}_{\mathcal{R}}$ and $\mathsf{Sign}_{\mathcal{R}}$.

Definition 3.6 (Class-hiding message space)**.** *An SPS-EQ scheme* SPS-EQ *over* $\mathbb{G}_i$ *has a* class-hiding *message space if for all* $\ell > 1$ *and all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that*

$$\Pr\left[\begin{array}{l|r} \mathsf{BG} \xleftarrow{R} \mathsf{BGGen}_{\mathcal{R}}(1^\kappa),\ b \xleftarrow{R} \{0,1\},\ M \xleftarrow{R} (\mathbb{G}_i^*)^\ell, & \\ M_0 \xleftarrow{R} [M]_{\mathcal{R}}, M_1 \xleftarrow{R} (\mathbb{G}_i^*)^\ell, & b^* = b \\ b^* \xleftarrow{R} \mathcal{A}(\mathsf{BG}, M, M_b) & \end{array}\right] - \frac{1}{2} \leq \epsilon(\kappa).$$

For the signatures, we require that signatures originating from $\mathsf{Sign}_{\mathcal{R}}$ are identically distributed to signatures output by $\mathsf{ChgRep}_{\mathcal{R}}$.

Definition 3.7 (Perfect adaptation of signatures)**.** *An SPS-EQ scheme* SPS-EQ *on* $(\mathbb{G}_i^*)^\ell$ perfectly adapts signatures *if for all tuples* $(\mathsf{sk}, \mathsf{pk}, M, \sigma, \mu)$ *with*

$$\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk}) = 1 \qquad \mathsf{Verify}_{\mathcal{R}}(M, \sigma, \mathsf{pk}) = 1 \qquad M \in (\mathbb{G}_i^*)^\ell \qquad \mu \in \mathbb{Z}_p^*$$

$\mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \mu, \mathsf{pk})$ *and* $(\mu M, \mathsf{Sign}_{\mathcal{R}}(\mu M, \mathsf{sk}))$ *are identically distributed.*

An even stronger property implying Definition 3.7 is the following one, which considers malicious keys and in a loose sense demands that signatures adapted by $\mathsf{ChgRep}_{\mathcal{R}}$ are uniform in the corresponding space of signatures:

Definition 3.8 (Perfect adaptation under malicious keys)**.** *An SPS-EQ scheme* SPS-EQ *on* $(\mathbb{G}_i^*)^\ell$ perfectly adapts signatures under malicious keys *if for all tuples* $(\mathsf{pk}, M, \sigma, \mu)$ *with*

$$\mathsf{Verify}_{\mathcal{R}}(M, \sigma, \mathsf{pk}) = 1 \qquad M \in (\mathbb{G}_i^*)^\ell \qquad \mu \in \mathbb{Z}_p^* \tag{3.1}$$

we have that $\mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \mu, \mathsf{pk})$ *outputs* $(\mu M, \sigma')$ *such that* σ' *is a random element in the space of signatures, conditioned on* $\mathsf{Verify}_{\mathcal{R}}(\mu M, \sigma', \mathsf{pk}) = 1$*.*

Later, in Proposition 3.10, we show that Definitions 3.6 and 3.7 imply Definition 3.5.

3.3 General Properties

The following proposition supports the intuition and ties the class-hiding property on the message space to the DDH assumption.

Proposition 3.9. *Let $\ell > 1$ and* SPS-EQ *be an SPS-EQ scheme on $(\mathbb{G}_i^*)^\ell$, then the message space $(\mathbb{G}_i^*)^\ell$ is class-hiding if and only if the DDH assumption holds in $\mathbb{G}_i$.*

Proof. W.l.o.g. we consider message space $(\mathbb{G}_1^*)^\ell$.

(1) We will show that class-hiding on $(\mathbb{G}_1^*)^\ell$ implies the DDH assumption in $\mathbb{G}_1$. Let $\mathcal{B}$ be a challenger against the class-hiding property on the message-space interacting with a DDH distinguisher $\mathcal{A}$ for $\mathbb{G}_1$. Initially, $\mathcal{B}$ is given a class-hiding message space instance (BG, M, M'). $\mathcal{B}$ then randomly selects two distinct indexes $i, j \in [\ell]$ starts $\mathcal{A}$ on (M_i, M_j, M'_i, M'_j). Eventually, $\mathcal{A}$ will output b' and then $\mathcal{B}$ will output $b^* \leftarrow b'$.

Observe that if $M' \in [M]_\mathcal{R}$, then there exists $\lambda \in \mathbb{Z}_p^*$ such that $\lambda M = M'$. Therefore, $(M_i, M_j, M'_i, M'_j) = (m_i P, m_j P, \lambda m_i P, \lambda m_j P)$ is a valid DDH tuple in $\mathbb{G}_1$ in this case. Finally, we have to consider the case of false positives, i.e., the case that $M' \notin [M]_\mathcal{R}$ but the input given to $\mathcal{A}$ constitutes a valid DDH tuple in $\mathbb{G}_1$. The probability of this event to occur is $O(\frac{1}{(p-1)^3})$ and thus negligible.

(2) We show that the DDH assumption on $\mathbb{G}_1$ implies a class-hiding message-space $(\mathbb{G}_1^*)^\ell$. Let us parametrize the game from Definition 3.6 with bit b and define Game_b to be the according version of the game. More precisely, $\mathcal{A}$ is given $(\mathsf{BG}, M, M_0 \xleftarrow{R} [M]_\mathcal{R})$ in Game_0 and $(\mathsf{BG}, M, M_1 \xleftarrow{R} (\mathbb{G}_1^*)^\ell)$ in Game_1, respectively.

Let $M = (M_i)_{i\in[\ell]}$. We next define a game Game'_j for all $j \in [\ell]$, where $\mu \xleftarrow{R} \mathbb{Z}_p^*$ and $R_{j+1}, \ldots, R_\ell \xleftarrow{R} \mathbb{G}_1^*$ and $\mathcal{A}$ is run on BG, M and

$$M' \coloneqq (\mu M_1, \ldots, \mu M_j, R_{j+1}, \ldots, R_\ell).$$

Note that by definition $\mathsf{Game}'_1 = \mathsf{Game}_0$ and $\mathsf{Game}'_\ell = \mathsf{Game}_1$.

Thus, if there exists an adversary that distinguishes Game_0 from Game_1 with probability $\epsilon(\kappa)$ then there must exist an index $j \in [\ell]$ such that the adversary distinguishes Game'_{j-1} from Game'_j with probability $\frac{1}{\ell-1}\epsilon(\kappa)$, which is non-negligible if $\epsilon(\kappa)$ is non-negligible. We show how to construct a DDH distinguisher from a distinguisher between Game'_{j-1} and Game'_j.

Given a DDH instance $(\mathsf{BG}, rP, sP, tP)$, we simulate the following game for the adversary. To do so, we pick $(m_i)_{i\in[\ell]} \xleftarrow{R} \mathbb{Z}_p^*$ and set

$$M \leftarrow \big(m_1 P, \ldots, m_{j-1} P, m_j(rP), m_{j+1} P, \ldots, m_\ell P\big). \tag{3.2}$$

Then, we sample $R_{j+1}, \ldots, R_\ell \xleftarrow{R} \mathbb{G}_1^*$, set

$$M' \leftarrow \big(m_1(sP), \ldots, m_{j-1}(sP), m_j(tP), R_{j+1}, \ldots, R_\ell\big) \tag{3.3}$$

and run $\mathcal{A}$ on (BG, M, M'). If $(\mathsf{BG}, rP, sP, tP)$ is a real DDH instance (i.e., $t = rs$) then the first j elements in (3.3) are s-multiples of the first j elements in (3.2), and we have thus simulated Game'_j. If t is random then so is the jth element in (3.3) and we have simulated Game'_{j-1}. Moreover, the simulation is perfect with overwhelming probability. The values r, s, t are drawn uniformly from $\mathbb{Z}_p$, whereas M and M' are supposed to be elements of $(\mathbb{G}_1^*)^\ell$. Therefore, it can happen only with negligible probability that $M \in \mathbb{G}_1^\ell \setminus (\mathbb{G}_1^*)^\ell$ and/or $M' \in \mathbb{G}_1^\ell \setminus (\mathbb{G}_1^*)^\ell$. Hence, any adversary distinguishing Game'_{j-1} from Game'_j thus breaks the DDH assumption. □

Proposition 3.10. *Let $\ell > 1$ and* SPS-EQ *be an SPS-EQ scheme on $(\mathbb{G}_i^*)^\ell$. If the message space of* SPS-EQ *is class-hiding and* SPS-EQ *perfectly adapts signatures, then* SPS-EQ *is class-hiding.*

Proof (Sketch). This proof is to some extent similar to the proof of the previous proposition. We will now outline the differences.

Depending on length ℓ, several game changes are again necessary (as in the proof of Proposition 3.9). During the simulation of the oracles in each of these games, we embed (by choosing additional random scalars) the values P, rP of a DDH instance $(\mathsf{BG}, rP, sP, tP)$ into every answer of the $\mathcal{O}^{RM}$ oracle and the values sP, tP at respective vector positions into every answer of the $\mathcal{O}^{RoR}$ oracle. When simulating the $\mathcal{O}^{RoR}$ oracle, we recompute all signatures; by perfect adaption fresh signatures are distributed like adapted signatures. In doing so, we can simulate a change of representatives if the instance is valid and the presentation of a random message-signature pair in the respective game otherwise. □

Remark 3.11. Finally, let us investigate the possibility of SPS-EQ in the Type-1 and Type-2 pairing setting and implied lower bounds. For the message-space to be class-hiding the DDH assumption has to hold on $\mathbb{G}_i$. This excludes the Type-1 setting, while in a Type-2 setting the message space can only be $(\mathbb{G}_1^*)^\ell$.

3.4 Relations to SPS

We now show how *any* EUF-CMA-secure SPS-EQ scheme that signs equivalence classes of $(\mathbb{G}_i^*)^{\ell+1}$ with $\ell > 0$ can be turned into an EUF-CMA secure SPS scheme signing vectors of $(\mathbb{G}_i^*)^\ell$. (We note, however, that SPS schemes typically allow messages from $\mathbb{G}_1$ and/or $\mathbb{G}_2$, which is preferable when used in combination with GS proofs [GS08].)

The transformation is simple and works by simply embedding messages $M = (M_i)_{i\in[\ell]} \in (\mathbb{G}_i^*)^\ell$ into $(\mathbb{G}_i^*)^{\ell+1}$ as $M' = (M, P)$ and signing M'. To verify a signature σ on a message $M \in (\mathbb{G}_i^*)^\ell$ under key pk, one then checks whether $\mathsf{Verify}_{\mathcal{R}}((M, P), \sigma, \mathsf{pk}) = 1$. This modification restricts each class to a single

representative, namely the one with P as its last element; a procedure we call *normalization*.

For security, see that EUF-CMA of the SPS-EQ states that no adversary can produce a signature on a message from an unqueried class, which therefore straight-forwardly implies EUF-CMA of the resulting SPS scheme.

Moreover, from any SPS-EQ with perfect adaptation of signatures the above transformation yields a rerandomizable SPS scheme, since signatures can be rerandomized by running $\mathsf{ChgRep}_{\mathcal{R}}$ for $\mu = 1$ (Definition 3.7 guarantees that this outputs a random signature).

This implication further means that the lower bounds for SPS over Type-3 bilinear groups given by Abe et al. in [AGHO11] carry over to EUF-CMA-secure SPS-EQs: Any EUF-CMA-secure SPS scheme must use at least 2 PPEs for verification and must have at least 3 signature elements, which cannot be from the same group (bilateral). Moreover, applying the result from [AGO11], this means that the EUF-CMA security of optimally short SPS-EQ schemes (i.e., schemes having 3-element signatures) cannot be reduced to non-interactive assumptions.

In [AGOT14a], Abe et al. identified the following lower bounds for Type-2 SPS schemes with messages in $\mathbb{G}_1$: 2 PPEs for verification and 3 group elements for signatures. The above transformation converts an EUF-CMA-secure Type-2 SPS-EQ into a Type-2 SPS, hence, these optimality criteria apply to EUF-CMA-secure Type-2 SPS-EQ schemes as well.

3.5 Constructions

In this section, we start by discussing an SPS-EQ construction secure in the GGM; then we show how to build a standard-model SPS-EQ from it.

3.5.1 A Generic-Group-Model Construction

In Scheme 2 we present our SPS-EQ construction from [FHS14] for message space $(\mathbb{G}_1^*)^\ell$. (One can construct a scheme for message space $(\mathbb{G}_2^*)^\ell$ by swapping the group memberships of all involved elements and adapting all computations accordingly.) Its signatures are constant-size (comprised of two $\mathbb{G}_1$ elements and one $\mathbb{G}_2$ element) and public keys consist of ℓ $\mathbb{G}_2$-elements. Moreover, verification requires only two PPEs. We first state the security of the signature scheme; the proofs will be given subsequently.

Security of the Construction

Now, we state the security of the signature scheme.

Theorem 3.12. *The SPS-EQ scheme in Scheme 2 is correct.*

Proof. We have to show that for all $\kappa \in \mathbb{N}$, all $\ell > 1$, all choices of bilinear groups $\mathsf{BG} \xleftarrow{R} \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$, all choices of key pairs $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^\ell)$,

Scheme 2 An EUF-CMA secure SPS-EQ scheme.

$\mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$: Given a security parameter 1^κ, output $\mathsf{BG} \xleftarrow{R} \mathsf{BGGen}(1^\kappa)$.

$\mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^\ell)$: Given a bilinear-group description BG and vector length $\ell > 1$ (in unary), choose $(x_i)_{i\in[\ell]} \xleftarrow{R} (\mathbb{Z}_p^*)^\ell$, set the secret key as $\mathsf{sk} \leftarrow (x_i)_{i\in[\ell]}$, compute the public key $\mathsf{pk} \leftarrow (\hat{X}_i)_{i\in[\ell]} = (x_i\hat{P})_{i\in[\ell]}$ and output $(\mathsf{sk}, \mathsf{pk})$.

$\mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk}; y)$: Given a message $M = (M_i)_{i\in[\ell]} \in (\mathbb{G}_1^*)^\ell$ defining equivalence class $[M]_{\mathcal{R}}$, a secret key $\mathsf{sk} = (x_i)_{i\in[\ell]} \in (\mathbb{Z}_p^*)^\ell$ and randomness $y \in \mathbb{Z}_p^*$; return $\sigma = (Z, Y, \hat{Y})$:

$$Z \leftarrow y \sum_{i\in[\ell]} x_i M_i \qquad Y \leftarrow \tfrac{1}{y} P \qquad \hat{Y} \leftarrow \tfrac{1}{y}\hat{P}$$

$\mathsf{Verify}_{\mathcal{R}}(M, \sigma, \mathsf{pk})$: Given a message $M = (M_i)_{i\in[\ell]} \in (\mathbb{G}_1^*)^\ell$, a signature $\sigma = (Z, Y, \hat{Y}) \in \mathbb{G}_1 \times \mathbb{G}_1^* \times \mathbb{G}_2^*$ and public key $\mathsf{pk} = (\hat{X}_i)_{i\in[\ell]} \in (\mathbb{G}_2^*)^\ell$, output 1 if the following holds and 0 otherwise.

$$\prod_{i\in[\ell]} e(M_i, \hat{X}_i) = e(Z, \hat{Y}) \qquad \wedge \qquad e(Y, \hat{P}) = e(P, \hat{Y})$$

$\mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \mu, \mathsf{pk}; \psi)$: Given a message $M = (M_i)_{i\in[\ell]} \in (\mathbb{G}_1^*)^\ell$ defining equivalence class $[M]_{\mathcal{R}}$, a signature $\sigma = (Z, Y, \hat{Y}) \in \mathbb{G}_1 \times \mathbb{G}_1^* \times \mathbb{G}_2$, $\mu \in \mathbb{Z}_p^*$, public key $\mathsf{pk} \in (\mathbb{G}_2^*)^\ell$ and randomness $\psi \in \mathbb{Z}_p^*$, return $\perp$ if $\mathsf{Verify}_{\mathcal{R}}(M, \sigma, \mathsf{pk}) = 0$. Otherwise, return $(\mu \cdot M, \sigma')$ with $\sigma' \leftarrow (\psi\mu Z, \frac{1}{\psi} Y, \frac{1}{\psi}\hat{Y})$.

$\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk})$: Given $\mathsf{sk} = (x_i)_{i\in[\ell]} \in (\mathbb{Z}_p^*)^\ell$ and $\mathsf{pk} = (\hat{X}_i)_{i\in[\ell]} \in (\mathbb{G}_2^*)^\ell$, output 1 if $x_i\hat{P} = \hat{X}_i \ \forall i \in [\ell]$ and 0 otherwise.

all $M \in (\mathbb{G}_1^*)^\ell$ and all $\mu \in \mathbb{Z}_p^*$ the following holds:

$$\begin{aligned}
&\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk}) = 1 \quad \wedge \\
&\mathsf{Verify}_{\mathcal{R}}\big(M, \mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk}), \mathsf{pk};\, y\big) = 1 \quad \forall\, y \in \mathbb{Z}_p^* \quad \wedge \\
&\mathsf{Verify}_{\mathcal{R}}\big(\mathsf{ChgRep}_{\mathcal{R}}(M, \mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk};\, y), \mu, \mathsf{pk};\, \psi), \mathsf{pk}\big) = 1 \quad \forall\, y, \psi \in \mathbb{Z}_p^*.
\end{aligned}$$

$\mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^\ell)$ returns $\mathsf{sk} \leftarrow (x_i)_{i\in[\ell]} \xleftarrow{R} (\mathbb{Z}_p^*)^\ell$ and $\mathsf{pk} \leftarrow (x_i\hat{P})_{i\in[\ell]}$, which shows the first equation.

$\mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk}; y)$ returns $Z = y\sum_{i\in[\ell]} x_i M_i$, $Y = \frac{1}{y} P$ and $\hat{Y} = \frac{1}{y}\hat{P}$. Plugging

this into the first relation in $\mathsf{Verify}_{\mathcal{R}}$, we get

$$e(Z,\hat{Y}) = e\big(y\textstyle\sum_{i\in[\ell]} x_i M_i, \frac{1}{y}\hat{P}\big) = e\big(\textstyle\sum_{i\in[\ell]} x_i M_i, \hat{P}\big)^{y\cdot\frac{1}{y}} = \prod_{i\in[\ell]} e(x_i M_i, \hat{P}) = \prod_{i\in[\ell]} e(M_i, \hat{X}_i).$$

Since $e(Y,\hat{P}) = e(\frac{1}{y}P,\hat{P}) = e(P,\frac{1}{y}\hat{P}) = e(P,\hat{Y})$, the second verification equation is also satisfied.

Finally, $\mathsf{ChgRep}_{\mathcal{R}}\big(M,(Z = y\sum_{i\in[\ell]} x_i M_i, Y = \frac{1}{y}P, \hat{Y} = \frac{1}{y}\hat{P}), \mu, \mathsf{pk}; \psi\big)$ outputs μM and

$$\hat{\sigma} = \big(\psi\mu Z,\ \tfrac{1}{\psi}Y,\ \tfrac{1}{\psi}\hat{Y}\big) = \big(\psi y\textstyle\sum_{i\in[\ell]} x_i\mu M_i,\ \frac{1}{\psi}\frac{1}{y}P,\ \frac{1}{\psi}\frac{1}{y}\hat{P}\big),$$

which is the same as $\mathsf{Sign}_{\mathcal{R}}(\mu M, \mathsf{sk}; (\psi y))$, and thus verifies by correctness of $\mathsf{Sign}_{\mathcal{R}}$. □

As already pointed out in Section 3.4, there is no reduction from the EUF-CMA security of any optimally short SPS-EQ scheme (that is, with 3-element signatures) to non-interactive assumptions. Since Scheme 2 fulfills this criterion, we are left to prove its security with a direct proof in the GGM (as also done by Abe et al. in [AGHO11, Lemma 1]).

Theorem 3.13. *Scheme 2 is EUF-CMA secure in the GGM for Type-3 bilinear groups groups.*

The proof is given in Appendix A.1.

Lemma 3.14. *Scheme 2 perfectly adapts signatures under malicious keys.*

Proof (Sketch). Let $\kappa > 0$, $(p,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,e,P,\hat{P}) \xleftarrow{R} \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$ and $\ell > 1$. For any $M \in (\mathbb{G}_1^*)^\ell$ and $\mathsf{pk} \in (\mathbb{G}_2^*)^\ell$, let $(x_i)_{i\in[\ell]}$ be such that $\mathsf{pk} = (x_i\hat{P})_{i\in[\ell]}$. A signature $\sigma = (Z,Y,\hat{Y}) \in \mathbb{G}_1 \times \mathbb{G}_1^* \times \mathbb{G}_2^*$ satisfying $\mathsf{Verify}_{\mathcal{R}}(M,\sigma,\mathsf{pk}) = 1$ must be of the form $(Z = y\sum x_i M_i, Y = \frac{1}{y}P, \hat{Y} = \frac{1}{y}\hat{P})$ for some $y \in \mathbb{Z}_p^*$. $\mathsf{ChgRep}_{\mathcal{R}}$ outputs $\sigma' = (y\psi\sum x_i\mu M_i, \frac{1}{y\psi}P, \frac{1}{y\psi}\hat{P})$, which is a random element in $\mathbb{G}_1 \times \mathbb{G}_1^* \times \mathbb{G}_2^*$ satisfying $\mathsf{Verify}_{\mathcal{R}}(M,\sigma',\mathsf{pk}) = 1$. □

Using the above lemma (which, in particular, implies perfect adaptation as given in Definition 3.7) and Proposition 3.10, we obtain the subsequent corollary:

Corollary 3.15. *Scheme 2 is class-hiding.*

3.5.2 A Standard-Model Construction

We will now present the standard-model SPS-EQ construction from [FHS15a]. Following the approach by Abe et al. [AGHO11], we construct from scheme SPS-EQ, given as Scheme 2, an SPS-EQ scheme SPS-EQ′, given as Scheme 3, and prove that it satisfies EUF-CMA and class-hiding, both under non-interactive

assumptions. Note that for this kind of construction it is not possible to achieve perfect adaptation of signatures (Definition 3.7).

The scheme for ℓ-length messages is simply Scheme 2 with message space $(\mathbb{G}_1^*)^{\ell+2}$, where before each signing two random group elements are appended to the message. Scheme 3 features constant-size signatures (4 $\mathbb{G}_1$+1 $\mathbb{G}_2$ elements),

Scheme 3 A standard-model SPS-EQ construction from Scheme 2.

$\mathsf{BGGen}'_{\mathcal{R}}(1^\kappa)$: Given a security parameter 1^κ, output $\mathsf{BG} \xleftarrow{R} \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$.

$\mathsf{KeyGen}'_{\mathcal{R}}(\mathsf{BG}, 1^\ell)$: Given a bilinear group BG and $\ell > 1$ (in unary), output $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^{\ell+2})$.

$\mathsf{Sign}'_{\mathcal{R}}(M, \mathsf{sk})$: Given a message $M = (M_i)_{i\in[\ell]} \in (\mathbb{G}_1^*)^\ell$ and a secret key sk, choose $(R_1, R_2) \xleftarrow{R} (\mathbb{G}_1^*)^2$, compute $\tau \xleftarrow{R} \mathsf{Sign}_{\mathcal{R}}((M, R_1, R_2), \mathsf{sk})$ and output $\sigma \leftarrow (\tau, R_1, R_2)$.

$\mathsf{Verify}'_{\mathcal{R}}(M, \sigma, \mathsf{pk})$: Given a message $M = (M_i)_{i\in[\ell]} \in (\mathbb{G}_1^*)^\ell$, a signature $\sigma \leftarrow (\tau, R_1, R_2)$ and a public key pk, return $\mathsf{Verify}_{\mathcal{R}}((M, R_1, R_2), \tau, \mathsf{pk})$.

$\mathsf{ChgRep}'_{\mathcal{R}}(M, \sigma, \mu, \mathsf{pk})$: Given a message $M = (M_i)_{i\in[\ell]} \in (\mathbb{G}_1^*)^\ell$, a signature $\sigma \leftarrow (\tau, R_1, R_2)$, a scalar $\mu \in \mathbb{Z}_p^*$ and a public key pk, run $((\tilde{M}, \tilde{R}_1, \tilde{R}_2), \tilde{\tau}) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}((M, R_1, R_2), \tau, \mu, \mathsf{pk})$ and output $(\tilde{M}, \tilde{\sigma})$ with $\tilde{\sigma} \leftarrow (\tilde{\tau}, \tilde{R}_1, \tilde{R}_2)$ (or $\perp$ if $\mathsf{ChgRep}_{\mathcal{R}}$ output $\perp$).

$\mathsf{VKey}'_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk})$: Given a key pair $(\mathsf{sk}, \mathsf{pk})$, return $\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk})$.

has public keys of size $\ell + 2$ and still uses 2 PPEs for verification.

Security of Scheme 3

Unforgeability follows from a q-type assumption stating that Scheme 2 for $\ell = 2$ is secure against *random-message attacks*. (That is, no PPT adversary, given the public key and signatures on q random messages, can, with non-negligible probability, output a message-signature pair for an equivalence class that was not signed.) Class-hiding follows from class-hiding of Scheme 2.

In order to prove the EUF-CMA security of Scheme 3, we introduce the following non-interactive q-type assumption. It is derived directly from Scheme 2 for $\ell = 2$, essentially stating that Scheme 2 is secure against random-message attacks.

Assumption 3.16. *Given a bilinear group* $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$*, two group elements* $(\hat{Y}_1, \hat{Y}_2) \xleftarrow{R} (\mathbb{G}_2^*)^2$ *and* q *instances* $(A_{j1}, A_{j2}, B_j, C_j, \hat{C}_j) \in (\mathbb{G}_1^*)^2 \times \mathbb{G}_1 \times \mathbb{G}_1^* \times \mathbb{G}_2^*$ *such that for all* $j \in [q]$

$$e(A_{j1}, \hat{Y}_1) \cdot e(A_{j2}, \hat{Y}_2) = e(B_j, \hat{C}_j) \quad \wedge \quad e(C_j, \hat{P}) = e(P, \hat{C}_j),$$

holds, then it is hard to output $(A_1^*, A_2^*, B^*, C^*, \hat{C}^*) \in (\mathbb{G}_1^*)^2 \times \mathbb{G}_1 \times \mathbb{G}_1^* \times \mathbb{G}_2^*$ *such that* $(A_1^*, A_2^*) \neq k \cdot (A_{j1}, A_{j2})$ *for all* $k \in \mathbb{Z}_p^*$, $j \in [q]$, *and*

$$e(A_1^*, \hat{Y}_1) \cdot e(A_2^*, \hat{Y}_2) = e(B^*, \hat{C}^*) \quad \wedge \quad e(C^*, \hat{P}) = e(P, \hat{C}^*).$$

Proof. Theorem 3.13 implies that Assumption 3.16 holds in the GGM. When reconsidering the simulation error analysis in the proof of Theorem 3.13 in Appendix A.1, we see that the degree of all involved polynomials is constant. Therefore, a generic adversary making $O(q)$ queries to the group oracles has probability $O(q^2/p)$ of breaking the assumption and thus the assumption reaches the optimal simulation error bound. □

We are now going to prove the unforgeability and the class-hiding property of Scheme 3. Correctness immediately follows from correctness of Scheme 2.

Theorem 3.17. *If Assumption 3.16 holds, then Scheme 3 is an EUF-CMA-secure SPS-EQ scheme.*

Proof. We assume that there is an efficient adversary $\mathcal{A}$ against the unforgeability of Scheme 3 that makes q' signing queries and use $\mathcal{A}$ to build an efficient adversary $\mathcal{B}$ against Assumption 3.16 for $q = q'$.

$\mathcal{B}$ is given $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$, $(\hat{V}_1, \hat{V}_2) \in (\mathbb{G}_2^*)^2$ and instances $(N_{j1}, N_{j2}, Z_j, Y_j, \hat{Y}_j)$ for $j \in [q]$. For all $i \in [\ell]$, $\mathcal{B}$ chooses $a_i, b_i \xleftarrow{R} \mathbb{Z}_p$ and computes $\hat{X}_i \leftarrow a_i \hat{V}_1 + b_i \hat{V}_2$. It sets $\hat{X}_{\ell+1} \leftarrow \hat{V}_1$, $\hat{X}_{\ell+2} \leftarrow \hat{V}_2$, $\mathsf{pk} \leftarrow (\hat{X}_i)_{i \in [\ell+2]}$ and runs $\mathcal{A}^{\mathsf{Sign}(\cdot, \mathsf{sk})}(\mathsf{pk})$. With overwhelming probability all elements $X_1, \ldots, X_{\ell+2}$ are non-trivial, in which case pk is distributed as a key in Scheme 3.

Next, $\mathcal{B}$ simulates $\mathcal{A}$'s queries to its signing oracle as follows. On the jth signing query for message $M_j = (M_{ji})_{i \in [\ell]} \in (\mathbb{G}_1^*)^\ell$, $\mathcal{B}$ computes

$$R_{j1} \leftarrow N_{j1} - \textstyle\sum_{i \in [\ell]} a_i M_{ji} \quad \text{and} \quad R_{j2} \leftarrow N_{j2} - \textstyle\sum_{i \in [\ell]} b_i M_{ji} \tag{3.4}$$

and returns the signature $\sigma_j \leftarrow ((Z_j, Y_j, \hat{Y}_j), R_{j1}, R_{j2})$ to $\mathcal{A}$. Note that the elements R_{j1} and R_{j2} are random, since N_{j1} and N_{j2} from the instance are random. (There is a small simulation error, as N_{j1} is uniform in $\mathbb{G}_1^*$, whereas R_{j1} is uniformly random in $\mathbb{G}_1^* \setminus \{-\sum_{i \in [\ell]} a_i M_{ji}\}$, but this error is negligible.) Moreover, they perfectly mask the scalars a_i and b_i.

Observe that the simulated signature satisfies the first verification equation:

$$\prod_{i \in [\ell]} e(M_{ji}, \hat{X}_i)\, e(R_{j1}, \hat{X}_{\ell+1})\, e(R_{j2}, \hat{X}_{\ell+2}) =$$

$$\prod_{i \in [\ell]} e(M_{ji}, a_i \hat{V}_1 + b_i \hat{V}_2)\, e(N_{j1} - \textstyle\sum_{i \in [\ell]} a_i M_{ji}, \hat{V}_1)\, e(N_{j2} - \textstyle\sum_{i \in [\ell]} b_i M_{ji}, \hat{V}_2) =$$

$$\prod_{i \in [\ell]} e(M_{ji}, a_i \hat{V}_1) \prod_{i \in [\ell]} e(M_{ji}, b_i \hat{V}_2)\, e(N_{j1}, \hat{V}_1)\, e(N_{j2}, \hat{V}_2) \cdot$$

$$\cdot \prod_{i \in [\ell]} e(a_i M_{ji}, \hat{V}_1)^{-1} \prod_{i \in [\ell]} e(b_i M_{ji}, \hat{V}_2)^{-1} =$$

$$e(N_{j1}, \hat{V}_1)\, e(N_{j2}, \hat{V}_2) = e(Z_j, \hat{Y}_j).$$

Since $(Y_j, \hat{Y}_j)$ from the instance are uniformly random in $\mathbb{G}_1^* \times \mathbb{G}_2^*$ conditioned on $e(Y_j, \hat{P}) = e(P, \hat{Y}_j)$ and together with $(M_{ji})_{i\in[\ell]}$, (R_{j1}, R_{j2}) and $(\hat{X}_i)_{i\in[\ell+2]}$, they uniquely determine Z_j as per the above equation, this shows that $(Z_j, Y_j, \hat{Y}_j)$ is a correctly distributed Scheme-2 signature. Furthermore, with overwhelming probability we have that $R_{j1} \neq 0_{\mathbb{G}_1}$ and $R_{j2} \neq 0_{\mathbb{G}_1}$, in which case the signatures $\sigma_j = ((Z_j, Y_j, \hat{Y}_j), R_{j1}, R_{j2})$ are perfectly simulated.

If $\mathcal{A}$ outputs a forgery $(M^*, \sigma^*) = ((M_i^*)_{i\in[\ell]}, (Z^*, Y^*, \hat{Y}^*, R_1^*, R_2^*))$ then $\mathcal{B}$ computes

$$N_1^* \leftarrow R_1^* + \textstyle\sum_{i\in[\ell]} a_i M_i^* \quad \text{and} \quad N_2^* \leftarrow R_2^* + \sum_{i\in[\ell]} b_i M_i^* \tag{3.5}$$

and returns $(N_1^*, N_2^*, Z^*, Y^*, \hat{Y}^*)$.

In order to show that $\mathcal{B}$'s output breaks Assumption 3.16, we need to show the following: (1) $(N_1^*, N_2^*, Z^*, Y^*, \hat{Y}^*)$ satisfies the last pair of equations in Definition 3.16; (2) $(N_1^*, N_2^*) \in (\mathbb{G}_1^*)^2$ and (3) $(N_1^*, N_2^*) \neq \mu \cdot (N_{j1}, N_{j2})$ for all $\mu \in \mathbb{Z}_p^*$, $j \in [q]$.

(1) We have:

$$\begin{aligned} e(N_1^*, \hat{V}_1)\, e(N_2^*, \hat{V}_2) &\overset{(3.5)}{=} e(\sum_{i\in[\ell]} a_i M_i^*, \hat{V}_1)\, e(\sum_{i\in[\ell]} b_i M_i^*, \hat{V}_2)\, e(R_1^*, \hat{V}_1)\, e(R_2^*, \hat{V}_2) \\ &= \Big(\prod_{i\in[\ell]} e(M_i^*, a_i \hat{V}_i) e(M_i^*, b_i \hat{V}_2) \Big) e(R_1^*, \hat{V}_1)\, e(R_2^*, \hat{V}_2) \\ &= \prod_{i\in[\ell]} e(M_i^*, \hat{X}_i)\, e(R_1^*, \hat{V}_1)\, e(R_2^*, \hat{V}_2) = e(Z^*, \hat{Y}^*), \end{aligned}$$

where the last equation follows from $\mathcal{A}$ outputting a valid signature. Since for the same reason, $e(Y^*, \hat{P}) = e(P, \hat{Y}^*)$, we have that $\mathcal{B}$'s output satisfies the required equations. (Note also that $Y^* \neq 0$ and $\hat{Y}^* \neq 0$ when $\mathcal{A}$'s output is valid.)

(2) The only information about $(a_i)_{i\in[\ell]}$ and $(b_i)_{i\in[\ell]}$ revealed to $\mathcal{A}$ is

$$x_i = a_i \cdot v_1 + b_i \cdot v_2, \tag{3.6}$$

where x_i, v_1 and v_2 are s.t. $\hat{X}_i = x_i \hat{P}$, $\hat{V}_1 = v_1 \hat{P}$ and $\hat{V}_2 = v_2 \hat{P}$, for all $i \in [\ell]$.

Since $M_i^* \neq 0$ for all $i \in [\ell]$, the probability that either $N_1^* = R_1^* + \sum_{i\in[\ell]} a_i M_i^* = 0$ or $N_2^* = R_2^* + \sum_{i\in[\ell]} b_i M_i^* = 0$ is therefore negligible.

3) Since M^* is a valid forgery, we have that for all $\mu \in \mathbb{Z}_p^*$ and $j \in [q]$: $M^* \neq \mu \cdot M_j$. The reduction could however fail if for some $\mu \in \mathbb{Z}_p$ and $j \in [q]$, we had $(N_1^*, N_2^*) = \mu \cdot (N_{j1}, N_{j2})$, that is

$$n_1^* \cdot n_{j2} = n_2^* \cdot n_{j1}, \tag{3.7}$$

where we let lower-case letters denote the logarithms of the corresponding upper-case letters to the basis P. We now show that even for an unbounded adversary, the probability that this happens is negligible.

$\mathcal{A}$ has no information about $(a_i)_{i\in[\ell]}$, however, by (3.6), each a_i determines b_i as

$$b_i = x_i v_2^{-1} - v_1 v_2^{-1} a_i.$$

Together with (3.4) and (3.5) this means that Equation (3.7) can be written as

$$\begin{aligned}&\big(r_1^* + \textstyle\sum_{i\in[\ell]} a_i m_i^*\big) \cdot \big(r_{j2} + \textstyle\sum_{i\in[\ell]} x_i v_2^{-1} m_{ji} - v_1 v_2^{-1} \sum_{i\in[\ell]} a_i m_{ji}\big)\\ &\quad = \big(r_2^* + \textstyle\sum_{i\in[\ell]} x_i v_2^{-1} m_i^* - v_1 v_2^{-1} \sum_{i\in[\ell]} a_i m_i^*\big) \cdot \big(r_{j1} + \textstyle\sum_{i\in[\ell]} a_i m_{ji}\big).\end{aligned}$$

This can be rewritten as (note that the terms containing products of a_i's cancel):

$$\begin{aligned}&\sum_{i\in[\ell]} \Big(- r_1^* v_1 v_2^{-1} m_{ji} + r_{j2} m_i^* + \textstyle\sum_{k\in[\ell]} x_k v_2^{-1} m_{jk} m_i^*\\ &\qquad + r_{j1} v_1 v_2^{-1} m_i^* - r_2^* m_{ji} - \textstyle\sum_{k\in[\ell]} x_k v_2^{-1} m_{ji} m_k^*\Big) a_i\\ &\quad = - r_1^* \big(r_{j2} + \textstyle\sum_{i\in[\ell]} x_i v_2^{-1} m_{ji}\big) + \big(r_2^* + \textstyle\sum_{i\in[\ell]} x_i v_2^{-1} m_i^*\big) r_{j1}.\end{aligned}$$

Since $\mathcal{A}$ has no knowledge of the a_i's, $\mathcal{A}$ can only make the equation be satisfied with non-negligible probability by setting all coefficients of the a_i's to 0. That is, for all $i \in [\ell]$:

$$\begin{aligned}\big(r_{j2} + r_{j1} v_1 v_2^{-1} + \textstyle\sum_k x_k v_2^{-1} m_{jk} - x_i v_2^{-1} m_{ji}\big) m_i^* - \textstyle\sum_{k\neq i} x_k v_2^{-1} m_{ji} m_k^*&\\ = \big(r_1^* v_1 v_2^{-1} + r_2^*\big) m_{ji}.&\quad (3.8)\end{aligned}$$

We now argue that the above system of ℓ linear equations in the variables $(m_1^*, \ldots, m_\ell^*)$ is regular with overwhelming probability. Indeed, $\mathcal{A}$ can choose the m_{jk}'s contained in the coefficients to its liking. However, it only learns r_{j2} afterwards, which is uniformly random (determined via the random N_{j2} from the instance). Thus, to the matrix determined by $\mathcal{A}$'s choices a (the same) random element is added to each entry in the diagonal; that is, a random multiple of the unity matrix I is added. It follows from the following claim that this makes the matrix regular with overwhelming probability.

Claim 3.18. *Let $A \in \mathbb{Z}_p^{\ell\times\ell}$. Then $A+\eta I$ for $\eta \xleftarrow{R} \mathbb{Z}_p$ is regular with overwhelming probability.*

Proof. Consider the Schur decomposition of A, that is, a regular matrix Q and an upper triangular matrix U, such that $A = QUQ^{-1}$. A is regular if and only if all diagonal elements of U are non-zero. $A + \eta I = Q(U + \eta I)Q^{-1}$ is regular if $(U+\eta I)$ has no zeros in the diagonal, which holds with overwhelming probability since the probability that $-\eta$ occurs in the diagonal of U is negligible. □

Let $r_1^*, r_2^*, m_{j1}, \ldots, m_{j\ell}$ be arbitrary. We then argue that the only assignment to $m^* = (m_1^*, \ldots, m_\ell^*)$ that satisfies the equation system in (3.8) is a multiple of m_j. This however means that the adversary did not win the unforgeability game.

Let $\lambda = (r_1^* v_1 v_2^{-1} + r_2^*)(r_{j2} + r_{j1} v_1 v_2^{-1})^{-1}$. Then $m_i^* \leftarrow \lambda m_{ji}$, for all $i \in [\ell]$, is a solution to the equation system in (3.8):

$$\begin{aligned}\big(r_{j2} + r_{j1} v_1 v_2^{-1} + \textstyle\sum_k x_k v_2^{-1} m_{jk} - x_i v_2^{-1} m_{ji}\big)\lambda m_{ji} - \textstyle\sum_{k \neq i} x_k v_2^{-1} m_{ji} \lambda m_{jk} \\ = \big(r_{j2} + r_{j1} v_1 v_2^{-1}\big)\lambda m_{ji} = \big(r_1^* v_1 v_2^{-1} + r_2^*\big) m_{ji}.\end{aligned}$$

Since the system is regular with overwhelming probability, this is the only solution, meaning in this case the adversary did not win. With overwhelming probability $\mathcal{B}$ thus returns a pair (N_1^*, N_2^*), which is not the multiple of any pair (N_{j1}, N_{j2}) from the given instance.

From an adversary $\mathcal{A}$ breaking unforgeability of Scheme 3, we have constructed an algorithm $\mathcal{B}$ which breaks Assumption 3.16 with almost the same probability; this completes the proof. □

Next, we prove Scheme 2 to be class-hiding.

Theorem 3.19. *If Scheme 2 is class-hiding, then Scheme 3 is class-hiding.*

Proof. We assume that there is an efficient adversary $\mathcal{A}$ against the class-hiding property of Scheme 3 with message length ℓ and use $\mathcal{A}$ to build an efficient adversary $\mathcal{B}$ against class-hiding of Scheme 2 with length $\ell + 2$.

$\mathcal{B}$ interacts with a class-hiding challenger $\mathcal{C}$, which creates the bilinear group BG and runs $\mathcal{B}$ on $(\mathsf{BG}, 1^{\ell+2})$. $\mathcal{B}$ runs $(\mathsf{st}_{\mathcal{A}}, \mathsf{sk} = (x_i)_{i\in[\ell+2]}, \mathsf{pk} = (\hat{X}_i)_{i\in[\ell+2]}) \leftarrow \mathcal{A}(\mathsf{BG}, 1^{\ell})$ and forwards this to $\mathcal{C}$. When $\mathcal{C}$ then runs $\mathcal{B}$ on $(\mathsf{st}_{\mathcal{A}}, \mathsf{sk}, \mathsf{pk})$, $\mathcal{B}$ runs $b^* \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{st}_{\mathcal{A}}, \mathsf{sk}, \mathsf{pk})$ and simulates $\mathcal{A}$'s oracles as follows.

On $\mathcal{A}$'s jth call to $\mathcal{O}^{RM}$, $\mathcal{B}$ calls its $\mathcal{O}^{RM}$ oracle to receive $M_j = (M_{ji})_{i\in[\ell+2]}$. $\mathcal{B}$ returns $(M_{ji})_{i\in[\ell]}$ to $\mathcal{A}$ and records $(M_{ji})_{i\in[\ell+2]}$.

When $\mathcal{A}$ calls the $\mathcal{O}^{RoR}$ oracle for message M_j, $\mathcal{B}$ looks for the first occurrence of M_j in its record, retrieves $(M_{ji})_{i\in[\ell+2]}$ and submits it to its $\mathcal{O}^{RoR}$ oracle. (If no entry in $\mathcal{B}$'s record starts with $(M_{j1}, \ldots, M_{j\ell})$ then $\mathcal{B}$ returns $\perp$.) Upon receiving $(M' = (M'_i)_{i\in[\ell+2]}, \sigma')$, $\mathcal{B}$ returns $((M'_i)_{i\in[\ell]}, (\sigma', M'_{\ell+1}, M'_{\ell+2}))$ to $\mathcal{A}$. Finally, $\mathcal{B}$ forwards $\mathcal{A}$'s output b^* to $\mathcal{C}$.

The simulation is perfect: On $\mathcal{A}$'s first valid call $M_j = (M_{ji})_{i\in[\ell]}$ to $\mathcal{O}^{RoR}$, it receives $M = M_j$ and $\sigma = (\sigma', M_{\ell+1}, M_{\ell+2})$, which is distributed the same way as $(M, \mathsf{Sign}'_{\mathcal{R}}(M, \mathsf{sk}))$, since $M_{\ell+1}, M_{\ell+2}$ are uniformly random elements (picked by $\mathcal{O}^{RM}$) and σ is a signature on $(M_i)_{i\in[\ell+2]}$, computed by $\mathcal{O}^{RoR}$.

Moreover, if $\mathcal{C}$'s bit $b = 0$ then at all further queries of M to $\mathcal{O}^{RoR}$, $\mathcal{B}$ receives $((M'_i)_{i\in[\ell+2]}, \sigma') \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}((M_i)_{i\in[\ell+2]}, \sigma, \mu, \mathsf{pk})$ for $\mu \xleftarrow{R} \mathbb{Z}_p^*$, and sends $((M'_i)_{i\in[\ell]}, (\sigma', M'_{\ell+1}, M'_{\ell+2}))$ to $\mathcal{A}$, which has the same distribution as the output of $\mathsf{ChgRep}'_{\mathcal{R}}((M_i)_{i\in[\ell]}, (\sigma, M_{i+1}, M_{i+2}), \mu, \mathsf{pk})$.

Finally, if $\mathcal{C}$'s bit $b = 1$ then at all further queries of M to $\mathcal{O}^{RoR}$, $\mathcal{B}$ receives $((R_i)_{i\in[\ell+2]}, \sigma')$ where $R_i \xleftarrow{R} (\mathbb{G}_i^*)^{\ell+2}$ and $\sigma' \xleftarrow{R} \mathsf{Sign}_{\mathcal{R}}(R, \mathsf{sk})$, and returns $((R_i)_{i\in[\ell]}, (\sigma', R_{\ell+1}, R_{\ell+2}))$ to $\mathcal{A}$, which is distributed the same way as $R \xleftarrow{R} (\mathbb{G}_i^*)^{\ell}$ and $\mathsf{Sign}'_{\mathcal{R}}((R_i)_{i\in[\ell]}, \mathsf{sk})$, and thus what $\mathcal{A}$ expects to receive. $\mathcal{B}$ thus wins the class-hiding game with the same probability as $\mathcal{A}$ does. □

Remark 3.20. Observe that Scheme 3 does not perfectly adapt signatures (Definition 3.7). The reason for this are the signature values R_1, R_2 whose common ratio is determined by the randomness initially used within $\mathsf{Sign}_{\mathcal{R}}$ and which remains the same after running $\mathsf{ChgRep}_{\mathcal{R}}$.

Deriving an SPS Scheme

Applying the transformation from Section 3.4 to Scheme 2, we obtain a perfectly rerandomizable SPS scheme in Type-3 groups with constant-size signatures of unilateral length-ℓ message vectors and public keys of size $\ell + 1$. Scheme 2 is optimal as it only uses 2 PPEs and its signatures consist of 3 bilateral group elements.

Applying our transformation to Scheme 3 yields a new standard-model SPS construction for unilateral length-ℓ message vectors in Type-3 groups. It has constant-size signatures (4 $\mathbb{G}_1 + 1$ $\mathbb{G}_2$ elements), a public key of size $\ell + 3$ and uses 2 PPEs for verification; it is therefore almost as efficient as the best known direct SPS construction from non-interactive assumptions in [AGHO11], whose signatures consist of 3 $\mathbb{G}_1 + 1$ $\mathbb{G}_2$ elements. Scheme 3 is partially rerandomizable [AFG+10], while the scheme in [AGHO11] is not.

3.6 Black-Box Separation of SPS-EQ from Non-Interactive Assumptions

In this section, we will show that it is impossible to base the EUF-CMA security of malicious-key perfectly adapting SPS-EQ schemes (Definition 3.8) on non-interactive assumptions, if the DDH assumption holds on the underlying group $\mathbb{G}_i$. In doing so, we use a meta-reduction technique that treats non-interactive hard problems (in the bilinear-group setting) as black-box and plays them off against the DDH assumption.

At this point, it must be made clear that this does not pose any serious problem to the notion of SPS-EQ and neither to its applications. In fact, it provides us with important evidence concerning the construction of standard-model SPS-EQs. In order to bypass our impossibility result, we will, then, propose a weaker unforgeability game in a one-more-forgery fashion: In order to win the game, an adversary must output $k+1$ valid, distinct and normalized message-signature pairs after having queried only k (unnormalized) messages to the signing oracle. This allows the reduction to efficiently distinguish between different classes. Most importantly, this notion is sufficient for all our use cases (that is, for blind signatures, multi-show ABCs and VESs), while the stronger notion is still useful if we do not require any indistinguishability on the message space, to prove schemes in the GGM or to construct schemes that are perfectly adapting (Definition 3.7; which is still sufficient for many use cases, e.g., for multi-show ABCs as in Chapter 7), but not perfectly adapting under malicious keys (Definition 3.8; which is required for blind signatures, cf. Chapter 4).

3.6.1 Hard Non-Interactive Problems

For the black-box separation of (certain) SPS-EQs from non-interactive assumptions, we need a formalization of hard non-interactive problems in the bilinear group setting. The main difference to Definitions 2.4 and 2.5 is that we explicitly take a bilinear-group generation algorithm into account.

Falsifiability is vital for our separation result (cf. Section 2.2.1), as the decision to the DDH output by the meta-reduction requires an efficient algorithm verifying the solution output by the reduction. Typical examples for problems in this setting that allow for public verification of solutions are extraction problems (a subclass of search problems), such as the discrete-logarithm problem (DLP), whereas examples for the other type are decision problems (e.g., DDH, DLIN) or certain search problems such as the CDH problem.

Definition 3.21 (Non-interactive problem)**.** *A* non-interactive problem (in the bilinear-group setting) P *consists of the following PPT algorithms:*

$\mathsf{BGGen}(1^\kappa)$: *A probabilistic algorithm that takes input a security parameter* 1^κ*. It outputs a bilinear group* BG*.*

$\mathsf{IGen}(\mathsf{BG}; r)$: *A probabilistic algorithm that takes input a bilinear group* BG *(and has access to a random tape* $r \in \{0,1\}^*$*). It outputs an instance* y *of* P*, where* y *includes* BG*.*

$\mathsf{V}(x, y, r)$: *A deterministic algorithm that takes input a value* x*, an instance* y *of* P *and randomness* $r \in \{0,1\}^*$ *such that* y *was generated using* r*. It outputs a decision bit indicating whether or not* x *is a solution of* y*.*

Definition 3.22 (Hard non-interactive problem)**.** *A non-interactive problem (in the bilinear-group setting)* P *is* hard*, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\,\mathsf{BG} \xleftarrow{R} \mathsf{BGGen}(1^\kappa),\ y \xleftarrow{R} \mathsf{IGen}(\mathsf{BG}; r),\ x \xleftarrow{R} \mathcal{A}(y)\ :\ \mathsf{V}(x, y, r) = 1\,\right] \leq \epsilon(\kappa).$$

3.6.2 The Separation Result

For our black-box separation, we use a meta-reduction technique, which plays off the combination of the DDH assumption on $\mathbb{G}_i$ and the perfect-adaptation-under-malicious-keys property against the EUF-CMA security. In our meta-reduction $\mathcal{M}$, the reduction $\mathcal{R}$ treats the adversary $\mathcal{A}$ as an oracle and the meta-reduction will use the reduction as signing oracle by applying a rewinding technique [Cor02]. Furthermore, we have to take into account that the meta-reduction may be confronted with a potentially malicious public key (e.g., generated by a simulator), while the meta-reduction must still be capable of deriving a fresh signature on the so-obtained message. For this reason, we require the perfect-adaptation-under-malicious-keys property (Definition 3.8).

We treat the underlying hard problem as black-box, that is, we do not make any assumption on it—except that it is hard, non-interactive and defined in a

bilinear-group setting (as given in Definition 2.4). Since we are considering an SPS type and, thus, also generic signers [AGHO11], the latter restriction does not seem to make any difference.

At first, we will state and prove the impossibility result for vanilla reductions, i.e., reductions that do not rewind and only run one instance of the adversary.

Theorem 3.23. *Let $\ell > 1$, $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$ be a bilinear group and* SPS-EQ *be an SPS-EQ scheme over $\mathbb{G}_i$. If the DDH assumption holds in $\mathbb{G}_i$ and the scheme is perfectly adapting under malicious keys, then there is no vanilla black-box reduction from its EUF-CMA security to a hard non-interactive problem in the bilinear-group setting given by* BG.

Proof. In the following, we will consider an (imaginary and not necessarily efficient) forger $\mathcal{F}$ that breaks the EUF-CMA security of the SPS-EQ scheme with probability $1/2$ as long as the challenger outputs valid signatures. $\mathcal{F}$ works as follows: When $\mathcal{F}$ receives the public key pk from the challenger, it picks $b \xleftarrow{R} \{0,1\}$ and queries a signature σ on one message $M \xleftarrow{R} (\mathbb{G}_i^*)^\ell$. If $\mathsf{Verify}_{\mathcal{R}}(M, \sigma, \mathsf{pk}) = 0$, $\mathcal{F}$ aborts. Else, if $b = 1$, $\mathcal{F}$ computes a signature σ^* on some new message $M^* \in (\mathbb{G}_i^*)^\ell \setminus [M]_{\mathcal{R}}$ and outputs (M^*, σ^*). For $b = 0$, $\mathcal{F}$ returns $(M^*, \sigma^*) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \mu, \mathsf{pk})$ by picking $\mu \xleftarrow{R} \mathbb{Z}_p^*$.

We will now show how to efficiently simulate $\mathcal{F}$ as part of the meta-reduction through a rewinding technique [Cor02], which can be applied in case of SPS-EQ schemes that are perfectly adapting under malicious keys; w.l.o.g. let $\ell = 2$.

We now describe our meta-reduction $\mathcal{M}$ and how it simulates the environment and the forger $\mathcal{F}$ for $\mathcal{R}$. $\mathcal{M}$ is given input a bilinear group $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$ with $\log_2 p = \lceil \kappa \rceil$ and a DDH instance $(\mathsf{BG}, rP, sP, tP)$ and runs the instance generator $y \xleftarrow{R} \mathsf{IGen}(\mathsf{BG}; \rho)$ of some hard non-interactive problem P (with random tape $\rho \in \{0,1\}^*$ made explicit). Then, $\mathcal{M}$ runs $\mathcal{R}$ on y. At some point, $\mathcal{R}$ will run $\mathcal{F}$ on pk, which is the public key pk determined by $\mathcal{R}$. Then, $\mathcal{M}$ simulates $\mathcal{F}$ as follows. $\mathcal{F}$ picks $m_1, m_2 \xleftarrow{R} \mathbb{Z}_p^*$ and submits $M' \leftarrow (m_1 P, m_2 rP)$ to $\mathcal{R}$'s signing oracle and gets in return a signature σ' on M'. If $\mathsf{Verify}_{\mathcal{R}}(M', \sigma', \mathsf{pk}) = 0$, $\mathcal{F}$ aborts. Otherwise, $\mathcal{M}$ rewinds $\mathcal{R}$ to the point before it runs $\mathcal{F}(\mathsf{pk})$ and lets $\mathcal{F}$ submit a new signature query for $M \leftarrow (m_1 sP, m_2 tP)$. When $\mathcal{R}$ responds with a signature σ on M, $\mathcal{F}$ outputs $(M^*, \sigma^*) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M', \sigma', \mu, \mathsf{pk})$ with $\mu \xleftarrow{R} \mathbb{Z}_p^*$ as a forgery to $\mathcal{B}$ unless $\mathsf{Verify}_{\mathcal{R}}(M, \sigma, \mathsf{pk}) = 0$ in which case $\mathcal{F}$ will abort. Eventually, $\mathcal{R}$ will then output a solution x for y to $\mathcal{M}$ and $\mathcal{M}$ will output $b^* \leftarrow 1 - \mathsf{V}(x, y, \rho)$ as decision bit for the DDH instance $(\mathsf{BG}, rP, sP, tP)$.

We analyze $\mathcal{M}$'s simulation of $\mathcal{F}$ for $\mathcal{R}$:

1. Only with negligible probability it can happen that $M \in \mathbb{G}_i^2 \setminus (\mathbb{G}_i^*)^2$ or that $M' \in \mathbb{G}_i^2 \setminus (\mathbb{G}_i^*)^2$.

2. For (M', σ') it holds that $\mathsf{Verify}_{\mathcal{R}}(M', \sigma', \mathsf{pk}) = 1$ and with overwhelming probability that $M' \in (\mathbb{G}_i^*)^2$. Then, the signature in $(M^*, \sigma^*) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M', \sigma', \mu, \mathsf{pk})$ for $\mu \xleftarrow{R} \mathbb{Z}_p^*$ is uniformly distributed in the set of all valid signatures on M^* under pk by the perfect adaptation under malicious keys property.

Now, if $(\mathsf{BG}, rP, sP, tP)$ is a valid DDH instance (i.e., $t = rs$), we simulate $\mathcal{F}$ for $b = 0$: $(M^*, \sigma^*) \in_R \mathsf{ChgRep}_{\mathcal{R}}(M', \sigma', \mu, \mathsf{pk})$ is not a forgery as $[(m_1 P, m_2 rP)]_{\mathcal{R}} = [M']_{\mathcal{R}} = [M^*]_{\mathcal{R}} = [M]_{\mathcal{R}} = [(m_1 sP, m_2 rsP)]_{\mathcal{R}}$. Else, if $(\mathsf{BG}, rP, sP, tP)$ is not a valid DDH instance and t is random (and, thus, independent of r and s), we simulate $\mathcal{F}$ for $b = 1$: $(M^*, \sigma^*) \in_R \mathsf{ChgRep}_{\mathcal{R}}(M', \sigma', \mu, \mathsf{pk})$ is a forgery as $[(m_1 P, m_2 rP)]_{\mathcal{R}} = [M']_{\mathcal{R}} = [M^*]_{\mathcal{R}} \neq [M]_{\mathcal{R}} = [(m_1 sP, m_2 tP)]_{\mathcal{R}}$. Therefore, after the rewind this simulates $\mathcal{F}$ perfectly unless $M \in \mathbb{G}_i^2 \setminus (\mathbb{G}_i^*)^2$ or $M' \in \mathbb{G}_i^2 \setminus (\mathbb{G}_i^*)^2$, which happens only with negligible probability.

If we are not dealing with a forgery (in case the DDH instance was valid), then $\mathcal{R}$ outputs x such that $\mathsf{V}(x, y, \rho) = 0$ and $\mathcal{M}$ outputs $b^* \leftarrow 1 - \mathsf{V}(x, y, \rho) = 1$ as decision bit to the DDH. Else, $\mathcal{R}$ outputs x such that $\mathsf{V}(x, y, \rho) = 1$, in which case $\mathcal{M}$ outputs $b^* \leftarrow 1 - \mathsf{V}(x, y, \rho) = 0$ as decision bit to the DDH. □

Now, we are going to extend this result to general reductions, that is, reductions that rewind the forger and/or run it multiple times (potentially with different public keys). Thereby, we follow a strategy similar to [Cor02].

Theorem 3.24. *Let the setting be the same as in Theorem 3.23, then there is no black-box reduction from its EUF-CMA security to a hard non-interactive problem in the bilinear-group setting given by* BG *that rewinds or sequentially runs a forger multiple times.*

Proof. We will show how the result from Theorem 3.23 can be extended to more general reductions, i.e., reductions that rewind or sequentially run multiple copies of a forger $\mathcal{F}'$, using the same techniques as before.

We start by assuming that the reduction $\mathcal{R}$ does not rewind the forger $\mathcal{F}'$, but sequentially runs $\mathcal{F}'$ r times, where $\mathcal{R}$ enters a new round each time an interaction with $\mathcal{F}'$ has completed. During each round, $\mathcal{F}'$ follows the same strategy as forger $\mathcal{F}$ in the proof of Theorem 3.23.

Next, we consider a reduction, which is allowed to rewind $\mathcal{F}'$ to some previous state S; or equivalently, a reduction $\mathcal{R}$ that restarts $\mathcal{F}'$ using the same random tape and the same input until state S has been reached. Restarting the forger means a transition of $\mathcal{R}$ from round $i - 1$ to round i. The strategy that $\mathcal{F}'$ pursues during each round is basically the same as the strategy of forger $\mathcal{F}$ in the proof of Theorem 3.23. $\mathcal{F}'$ takes the rewinding into account as follows. If $\mathcal{R}$ starts $\mathcal{F}'$ on the same public key in round i as in round $i - 1$, then $\mathcal{F}'$ will send the same signature query as in round $i - 1$. If $\mathcal{R}$ then returns the exact same signature as in round $i - 1$, $\mathcal{F}'$ will output in round i the exact same answer as in round $i - 1$ or an arbitrary output of $\mathsf{ChgRep}_{\mathcal{R}}(M', \sigma', \mu, \mathsf{pk})$ if $\mathcal{F}'$ was rewound in round $i - 1$ before returning the result. Otherwise, if $\mathcal{R}$ returns a different signature as in round $i - 1$, then $\mathcal{F}'$ will return in round i an arbitrary output of $\mathsf{ChgRep}_{\mathcal{R}}(M', \sigma', \mu, \mathsf{pk})$. Finally, if $\mathcal{R}$ has been restarted on a different public key, then $\mathcal{F}'$ will perform its queries and compute its result in round i independently of round $i - 1$.

In order to simulate the forger $\mathcal{F}'$, the meta-reduction uses the same technique as in the proof of Theorem 3.23. In doing so, it performs the rewinding of $\mathcal{R}$ in round i to the point before $\mathcal{R}$ runs $\mathcal{F}(\mathsf{pk})$ in round i. □

Remark 3.25. These results apply analogously to SPS-EQ schemes defined on more general projective equivalence relations as already pointed out in Remark 3.1. Again, considering equivalence classes, which are planes (and not lines), the DDH assumption on $\mathbb{G}_i$ would be replaced by the DLIN assumption to obtain indistinguishability on the message space. Likewise, the hardness of the DLIN assumption and an (analogous version of) perfect adaptation under malicious keys would give an analogous impossibility result.

3.6.3 Discussion and Alternative Unforgeability Notion

The results show that basing the EUF-CMA security (as in Definition 3.4) of a malicious-key perfectly-adapting SPS-EQ scheme (cf. Definition 3.8) on non-interactive assumptions is impossible. While Definition 3.8 is crucial to building blind signatures from SPS-EQ in the malicious-key model, investigating weaker unforgeability notions could lead us to SPS-EQ constructions whose unforgeability relies on non-interactive assumptions. In turn, this could allow obtaining blind signatures that are both unforgeable and blind under non-interactive assumptions.

This motivates the definition of a weaker unforgeability game, where the reduction is efficiently able to decide the class membership of the forgery. To this end, we introduce the following one-more-forgery unforgeability game, which is similar to the unforgeability notion of blind signatures (cf. Definition 4.3). It allows signing queries for arbitrary representatives of arbitrary classes, but, in the end, requires the adversary to output distinct, normalized representatives (i.e., representatives with generator P in one component; w.l.o.g. in the last) and corresponding signatures. The adversary wins the game if it is able to output $k+1$ message-signature pairs after having queried only k messages to the signing oracle.

Definition 3.26 (One-more unforgeability)**.** *An SPS-EQ scheme* $\mathsf{SPS\text{-}EQ}$ *on* $\mathbb{G}_i$ *is* one-more unforgeable *if for all* $\ell > 1$ *and all PPT adversaries* $\mathcal{A}$ *having access to a signing oracle* $\mathsf{Sign}_{\mathcal{R}}(\cdot, \mathsf{sk})$*, there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l|r} \mathsf{BG} \xleftarrow{R} \mathsf{BGGen}_{\mathcal{R}}(1^\kappa), & \\ (\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^\ell), & M_i \neq M_j \ \forall i, j \in [k+1],\ i \neq j \\ (M_i, \sigma_i)_{i \in [k+1]} & : \ \wedge \ \mathsf{Verify}_{\mathcal{R}}((M_i, P), \sigma_i, \mathsf{pk}) = 1 \\ \quad \xleftarrow{R} \mathcal{A}^{\mathsf{Sign}_{\mathcal{R}}(\cdot, \mathsf{sk})}(\mathsf{pk}) & \forall i \in [k+1] \end{array}\right] \leq \epsilon(\kappa),$$

where k *is the number of queries that* $\mathcal{A}$ *has issued to the signing oracle.*

4

Practically Efficient Round-Optimal Blind Signatures in the Standard Model

As we play with these shiny new toys,
how much are we trading off convenience over privacy and security?
— James Lyne, at #TED2013

Blind signatures (BSs) schemes [Cha82] allow a user to obtain a signature from a signer, in such a way that the signer cannot link the resulting message-signature pair to the signing process. They are a central cryptographic building block and have seen lots of attention since the 1980s. Not surprisingly, they also have a variety of applications; e.g., in e-cash, e-voting and anonymous authentication (one-show credential systems).

Important quality criteria of blind-signature schemes are the number of interactions required during the signing protocol and whether they assume random oracles (ROs) and/or a CRS set up by a TTP, which every party must be given access to; or if their proofs hold in the standard model. Schemes having only two moves of interaction are said to be *round-optimal* [Fis06]. Round-optimality, on the one hand, immediately implies concurrent security (which otherwise has to be taken into account separately; cf. [KZ06, HKKL07]) and is, on the other hand, a crucial criterion for a scheme's efficiency.

Based on SPS-EQs, we will now present a new round-optimal blind-signature scheme from [FHS15a], which we will later extend to a round-optimal partially blind-signature scheme. We achieve this with a new and conceptually simple approach that yields compact constructions which are efficient in terms of signature size, communication complexity, computational effort and key sizes.

Our schemes are surprisingly efficient and are secure in the standard model under interactive assumptions (which is the main caveat). Unlike previous schemes [GRS+11, GG14], our security proofs hold in the standard model without the need for complexity leveraging and non-uniformity. Our partially blind-signature scheme is—to the best of our knowledge—the first such construction secure in the standard-model. Finally, we will show how to build blind signatures on message vectors, which we will use subsequently to build one-show anonymous credentials in Section 7. This is the first such credential system in the vein of Brands relying on a blind-signature scheme with security proofs in the standard model.

Before giving the contribution, we state the background necessary on blind and partially blind signatures. The remaining parts of this chapter are based on joint work with Georg Fuchsbauer and Daniel Slamanig. The presented material is taken (mostly verbatim) from [FHS15a, FHS15b].

4.1 Blind Signatures: Definitions

A BS scheme is a two-party protocol, run between a user (or obtainer) and a signer (or issuer) satisfying the following definition:

Definition 4.1 (Blind signature (BS) scheme)**.** *A* BS *scheme* BS *consists of the following PPT algorithms:*

$\mathsf{KeyGen}(1^\kappa)$: *A probabilistic algorithm that takes input a security parameter* 1^κ*. It returns a key pair* $(\mathsf{sk}, \mathsf{pk})$ *(we assume that* pk *includes a description of the message space* $\mathcal{M}_{\mathsf{pk}}$*).*

$(\mathcal{U}(m, \mathsf{pk}), \mathcal{S}(\mathsf{sk}))$: *These algorithms are run by a user and a signer, who interact during execution.* $\mathcal{U}$ *is a probabilistic algorithm that takes input a message* $m \in \mathcal{M}_{\mathsf{pk}}$ *and a public key* pk*.* $\mathcal{S}$ *is a probabilistic algorithm that takes input a secret key* sk*. At the end of this protocol,* $\mathcal{U}$ *outputs* σ*, a signature on* m*, or* $\perp$ *if the interaction was not successful.*

$\mathsf{Verify}(m, \sigma, \mathsf{pk})$: *A deterministic algorithm that takes input a message-signature pair* (m, σ) *with* $m \in \mathcal{M}_{\mathsf{pk}}$ *and a public key* pk*. It returns* 1 *or* 0 *indicating whether or not* (m, σ) *is a valid message-signature pair under* pk*.*

A BS scheme BS must satisfy *correctness*, *unforgeability* and *blindness*.

Definition 4.2 (Correctness)**.** *A BS scheme* BS *is* correct *if for all security parameters* κ*, all choices of* $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa)$*, all messages* $m \in \mathcal{M}_{\mathsf{pk}}$ *it holds that*

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

Due to blindness, unforgeability is defined as a *one-more-forgery* game, i.e., the user wins the unforgeability game if she is able to output $k + 1$ distinct message-signature pairs after having queried only k times to the signer oracle.

Definition 4.3 (Unforgeability)**.** *A BS scheme* BS *is* unforgeable *if for all PPT adversaries* $\mathcal{A}$ *having access to a signer oracle, there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l|r} (\mathsf{sk},\mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa), & m_i^* \neq m_j^* \ \forall i,j \in [k+1], i \neq j \\ (m_i^*,\sigma_i^*)_{i\in[k+1]} \xleftarrow{R} \mathcal{A}^{(\cdot,\mathcal{S}(\mathsf{sk}))}(\mathsf{pk}) & : \quad \wedge\ \mathsf{Verify}(m_i^*,\sigma_i^*,\mathsf{pk}) = 1 \\ & \forall i \in [k+1] \end{array}\right] \leq \epsilon(\kappa),$$

where k *is the number of completed oracle interactions.*

There are several definitions of blindness; most prominently, the honest-signer [JLO97, AO00, CKW05] and the malicious-signer (or dishonest-signer) [ANN06, Oka06] model. In the former, the whole signer key pair $(\mathsf{sk},\mathsf{pk})$ is defined by the environment; whereas in the latter model the public key pk is adversarially generated. Apparently, this is a stronger, more desirable model, which, however, makes it more challenging (or under certain circumstances even impossible) to find security reductions. Evidence underlining this is given by Fischlin and Schröder in [FS10]. There they show that the unforgeability of a blind-signature scheme with blindness in the malicious-key model cannot be based on non-interactive hardness assumptions if (1) the scheme has 3 moves or less, (2) its blindness holds statistically and (3) from a transcript one can efficiently decide whether the interaction yielded a valid blind signature. They also give some evidence that their result applies to computationally blind schemes as well—under certain circumstances.

Besides these two notions, there is also a simulation-based blindness notion, for which Lindell showed in [Lin03] that such concurrently secure constructions are impossible in the standard model.

Definition 4.4 (Honest-signer blindness)**.** *A BS scheme* BS *is called* honest-signer blind *if for all PPT adversaries* $\mathcal{A}$ *having one-time access to two user oracles, there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l|r} b \xleftarrow{R} \{0,1\},\ (\mathsf{sk},\mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa), & \\ (\mathsf{st},m_0,m_1) \xleftarrow{R} \mathcal{A}(\mathsf{sk},\mathsf{pk}), & \\ \mathsf{st} \xleftarrow{R} \mathcal{A}^{(\mathcal{U}(m_b,\mathsf{pk}),\cdot)^1,(\mathcal{U}(m_{1-b},\mathsf{pk}),\cdot)^1}(\mathsf{st}), & \\ \textit{Let } \sigma_b \textit{ and } \sigma_{1-b} \textit{ be the resp. outputs of } \mathcal{U}, & : \ b^* = b \\ \textit{If } \sigma_0 = \bot \textit{ or } \sigma_1 = \bot \textit{ then } (\sigma_0,\sigma_1) \leftarrow (\bot,\bot), & \\ b^* \xleftarrow{R} \mathcal{A}(\mathsf{st},\sigma_0,\sigma_1) & \end{array}\right] - \frac{1}{2} \leq \epsilon(\kappa),$$

Definition 4.5 (Malicious-signer blindness [ANN06, Oka06])**.** *A BS scheme* BS *is called* (malicious-signer) blind *if for all PPT adversaries* $\mathcal{A}$ *having one-time access to two user oracles, there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l|r} b \xleftarrow{R} \{0,1\},\ (\mathsf{st},\mathsf{pk},m_0,m_1) \xleftarrow{R} \mathcal{A}(1^\kappa), & \\ \mathsf{st} \xleftarrow{R} \mathcal{A}^{(\mathcal{U}(m_b,\mathsf{pk}),\cdot)^1,(\mathcal{U}(m_{1-b},\mathsf{pk}),\cdot)^1}(\mathsf{st}), & \\ \textit{Let } \sigma_b \textit{ and } \sigma_{1-b} \textit{ be the resp. outputs of } \mathcal{U}, & : \ b^* = b \\ \textit{If } \sigma_0 = \bot \textit{ or } \sigma_1 = \bot \textit{ then } (\sigma_0,\sigma_1) \leftarrow (\bot,\bot), & \\ b^* \xleftarrow{R} \mathcal{A}(\mathsf{st},\sigma_0,\sigma_1) & \end{array}\right] - \frac{1}{2} \leq \epsilon(\kappa).$$

4.1.1 Partially Blind Signatures

Partially blind signatures (PBSs) are a generalization of BSs, which restrict the power of the user by including an additional piece of information (the *common information*), which is agreed upon between the user and the signer, into the signatures. We will now state the formal definitions:

Definition 4.6 (Partially blind signature (PBS) scheme)**.** *A* PBS *scheme* PBS *consists of the following PPT algorithms:*

$\mathsf{KeyGen}(1^\kappa)$: *A probabilistic algorithm that takes input a security parameter* 1^κ*. It returns a key pair* $(\mathsf{sk}, \mathsf{pk})$ *(we assume that* pk *includes a description of the message space* $\mathcal{M}_{\mathsf{pk}}$*).*

$(\mathcal{U}(m, \gamma, \mathsf{pk}), \mathcal{S}(\gamma, \mathsf{sk}))$: *These algorithms are run by a user and a signer, who interact during execution.* $\mathcal{U}$ *is a probabilistic algorithm that takes input a message* $m \in \mathcal{M}_{\mathsf{pk}}$*, common information* $\gamma \in \mathcal{M}_{\mathsf{pk}}$ *and a public key* pk*.* $\mathcal{S}$ *is a probabilistic algorithm that takes input common information* $\gamma \in \mathcal{M}_{\mathsf{pk}}$ *and a secret key* sk*. At the end of this protocol,* $\mathcal{U}$ *outputs* σ*, a signature on* (m, γ)*, or* $\perp$ *if the interaction was not successful.*

$\mathsf{Verify}(m, \gamma, \sigma, \mathsf{pk})$: *A deterministic algorithm that takes input a message-signature tuple* (m, γ, σ) *with* $m, \gamma \in \mathcal{M}_{\mathsf{pk}}$ *and a public key* pk*. It returns* 1 *or* 0 *indicating whether or not* (m, γ, σ) *is a valid message-signature pair under* pk*.*

A PBS scheme PBS is called *secure*, if it is *correct*, *unforgeable* and *partially blind*.

Definition 4.7 (Correctness)**.** *A PBS scheme* PBS *is* correct *if for all security parameters* κ*, all choices of* $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa)$*, all messages* $m \in \mathcal{M}_{\mathsf{pk}}$ *and all* $\gamma \in \mathcal{M}_{\mathsf{pk}}$ *it holds that*

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

Definition 4.8 (Unforgeability)**.** *A PBS scheme* PBS *scheme is* unforgeable*, if for all PPT adversaries* $\mathcal{A}$ *having access to a signer oracle, there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l|rl} (\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa), & m_i^* \neq m_j^* \ \forall i, j \in [k_{\gamma^*}+1], i \neq j & \\ (\gamma^*, (m_i^*, \sigma_i^*)_{i \in [k_{\gamma^*}+1]}) & : \quad \wedge \ \mathsf{Verify}(m_i^*, \gamma^*, \sigma_i^*, \mathsf{pk}) = 1 & \\ \xleftarrow{R} \mathcal{A}^{(\cdot, \mathcal{S}(\cdot, \mathsf{sk}))}(\mathsf{pk}) & \forall i \in [k_{\gamma^*}+1] & \end{array}\right] \leq \epsilon(\kappa),$$

where k_{γ^*} *is the number of completed oracle interactions involving* γ^**.*

We omit the definition of partial blindness for the honest-signer case and just state it for malicious signers [AO00]. (It is straightforward to derive the honest-signer partial blindness definition.)

Definition 4.9 (Malicious-signer partial blindness)**.** *A PBS scheme* PBS *is* (malicious-signer) partially blind*, if for all PPT adversaries* $\mathcal{A}$ *having one-time access to two user oracles, there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l} b \xleftarrow{R} \{0,1\},\ (\mathsf{st}, \mathsf{pk}, m_0, m_1, \gamma) \xleftarrow{R} \mathcal{A}(1^\kappa), \\ \mathsf{st} \xleftarrow{R} \mathcal{A}^{(\mathcal{U}(m_b,\gamma,\mathsf{pk}),\cdot)^1,(\mathcal{U}(m_{1-b},\gamma,\mathsf{pk}),\cdot)^1}(\mathsf{st}), \\ \textit{Let } \sigma_b \textit{ and } \sigma_{1-b} \textit{ be the resp. outputs of } \mathcal{U}, \\ \textit{If } \sigma_0 = \bot \textit{ or } \sigma_1 = \bot \textit{ then } (\sigma_0, \sigma_1) \leftarrow (\bot, \bot), \\ b^* \xleftarrow{R} \mathcal{A}(\mathsf{st}, \sigma_0, \sigma_1) \end{array} : b^* = b \right] - \frac{1}{2} \leq \epsilon(\kappa).$$

4.2 Building Blind Signatures from SPS-EQ

Our construction uses commitments to the messages and SPS-EQ to sign these commitments and to perform blinding and unblinding. Signing an equivalence class with an SPS-EQ scheme lets one derive a signature for arbitrary representatives of this class without knowing the private signing key. This concept provides an elegant way to realize a blind signing process as follows.

The signer's key contains an element Q under which the obtainer forms a Pedersen commitment $C = mP + rQ$ to the message m. (Since the commitment is perfectly hiding, the signer can be aware of q with $Q = qP$.) The obtainer then forms a vector (C, P), which can be seen as the canonical representative of equivalence class $[(C, P)]_{\mathcal{R}}$. Next, she picks $s \xleftarrow{R} \mathbb{Z}_p^*$ and moves (C, P) to a random representative (sC, sP), which hides C. She sends (sC, sP) to the signer and receives an SPS-EQ signature π on it, from which she can derive a signature σ to the original message (C, P), which she can then publish together with an opening of C. As verification will check validity of the SPS-EQ signature on a message ending with P, the unblinding is unambiguous.

Let us now discuss how the user opens the Pedersen commitment $C = mP + rQ$. Publishing (m, r) directly would break the blindness of the scheme (a signer could link a pair $M = (D, S)$, received during signing, to a signature by checking whether $D = mS + rqS$). We, therefore, define a tweaked opening, for which we include $\hat{Q} = q\hat{P}$ in addition to $Q = qP$ into the signer's public key. We define the opening as (m, rP), which can be checked via the pairing equation

$$e(C - mP, \hat{P}) = e(rP, \hat{Q}).$$

This opening is still computationally binding under the co-DHI assumption in $\mathbb{G}_1$ (in contrast to standard Pedersen commitments, which are binding under the DL assumption). Hiding of the commitment still holds unconditionally, and we will prove the constructed blind-signature scheme secure in the malicious-signer model without requiring a trusted setup.

Finally, we need to consider another technicality. In the malicious-key model the public key is fully controlled by the adversary. For Scheme 4 this means that also the bilinear group BG is under adversarial control. When reducing Assumption 4.11 to the blindness of Scheme 4, the bilinear group is, however, part of the problem instance. To guarantee that both bilinear groups are equal, we

require the bilinear-group generation algorithm $\mathsf{BGGen}_{\mathcal{R}}$ of the SPS-EQ scheme to be deterministic[1].

The scheme is presented as Scheme 4. (Note that for simplicity the blind signature contains $T = rQ$ instead of C.)

Scheme 4 A blind-signature scheme from SPS-EQ.

$\mathsf{KeyGen}(1^\kappa)$: Given a security parameter 1^κ, compute $\mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$, $(\mathsf{sk}, \mathsf{pk}_{\mathcal{R}}) \xleftarrow{R} \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^\ell)$ for $\ell = 2$, pick $q \xleftarrow{R} \mathbb{Z}_p^*$ and set $Q \leftarrow qP$, $\hat{Q} \leftarrow q\hat{P}$. Output $(\mathsf{sk}, \mathsf{pk} = (\mathsf{pk}_{\mathcal{R}}, Q, \hat{Q}))$.

$\mathcal{U}^{(1)}(m, \mathsf{pk})$: Given a message $m \in \mathbb{Z}_p$ and public key $\mathsf{pk} = (\mathsf{pk}_{\mathcal{R}}, Q, \hat{Q})$, compute $\mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$. If $Q = 0_{\mathbb{G}_1}$ or $e(Q, \hat{P}) \neq e(P, \hat{Q})$, return $\perp$; else choose $s \xleftarrow{R} \mathbb{Z}_p^*$ and $r \xleftarrow{R} \mathbb{Z}_p$ such that $mP + rQ \neq 0_{\mathbb{G}_1}$ and output

$$M \leftarrow (s(mP + rQ), sP) \qquad \mathsf{st} \leftarrow (\mathsf{BG}, \mathsf{pk}_{\mathcal{R}}, Q, M, r, s).$$

$\mathcal{S}(M, \mathsf{sk})$: Given $M \in (\mathbb{G}_1^*)^2$ and secret key sk, output $\pi \xleftarrow{R} \mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk})$.

$\mathcal{U}^{(2)}(\mathsf{st}, \pi)$: Given state st and π, parse st as $(\mathsf{BG}, \mathsf{pk}_{\mathcal{R}}, Q, M, r, s)$. If $\mathsf{Verify}_{\mathcal{R}}(M, \pi, \mathsf{pk}_{\mathcal{R}}) = 0$ then return $\perp$. Else, run

$$((mP + rQ, P), \sigma) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M, \pi, \frac{1}{s}, \mathsf{pk}_{\mathcal{R}}),$$

and output $\tau \leftarrow (\sigma, rP, rQ)$.

$\mathsf{Verify}(m, \tau, \mathsf{pk})$: Given message $m \in \mathbb{Z}_p^*$, blind signature $\tau = (\sigma, R, T)$ and $\mathsf{pk} = (\mathsf{pk}_{\mathcal{R}}, Q, \hat{Q})$, with $Q \neq 0_{\mathbb{G}_1}$ and $e(Q, \hat{P}) = e(P, \hat{Q})$, output 1 if the following holds and 0 otherwise.

$$\mathsf{Verify}_{\mathcal{R}}((mP + T, P), \sigma, \mathsf{pk}_{\mathcal{R}}) = 1 \qquad e(T, \hat{P}) = e(R, \hat{Q})$$

4.2.1 Security

We omit the proof of correctness, as it follows straightforwardly from inspection.

Theorem 4.10. *If the underlying SPS-EQ scheme* SPS-EQ *is EUF-CMA secure and the co-DHI assumption holds in* $\mathbb{G}_1$ *then Scheme 4 is unforgeable.*

The proof follows the intuition that a forger must either forge an SPS-EQ signature on a new commitment or open a commitment in two different ways.

[1] As already pointed out in Chapter 4, this is, e.g., the case for BN curves [BN06]; the most common choice for Type-3 pairings.

The reduction has—due to blindness—a security loss proportional to the number of signing queries.

Proof. To prove unforgeability of Scheme 4, we assume that there is an efficient adversary $\mathcal{A}$ winning the unforgeability game with non-negligible probability $\epsilon(\kappa)$. We then construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to either break the EUF-CMA security of SPS-EQ or to break the binding property of the underlying commitment scheme, that is, to break co-DHI in $\mathbb{G}_1$.

$\mathcal{B}$ first guesses $\mathcal{A}$'s strategy, i.e., the type of forgery $\mathcal{A}$ will conduct. We call a forgery *Type-1* if for $\mathcal{A}$'s output $(m_i, \tau_i = (\sigma_i, R_i, T_i))_{i\in[k+1]}$, we have $m_iP + T_i \neq m_jP + T_j$ for all $i \neq j$; otherwise we call it *Type-2*.

Type 1: $\mathcal{B}$ uses $\mathcal{A}$ to break the EUF-CMA security of the SPS-EQ scheme $\mathsf{SPS\text{-}EQ}$ with $\ell = 2$. $\mathcal{B}$ obtains $\mathsf{pk}_{\mathcal{R}}$ from its challenger $\mathcal{C}$, chooses $q \xleftarrow{R} \mathbb{Z}_p^*$, computes $(Q, \hat{Q}) \leftarrow q(P, \hat{P})$, sets $\mathsf{pk} \leftarrow (\mathsf{pk}_{\mathcal{R}}, Q, \hat{Q})$ and runs $\mathcal{A}(\mathsf{pk})$. Whenever $\mathcal{A}$ queries to the $(\cdot, \mathcal{S}(\mathsf{sk}))$ oracle and sends a blinded message M during the interaction, $\mathcal{B}$ queries $\mathcal{C}$'s SPS-EQ signing oracle $\mathsf{Sign}_{\mathcal{R}}(\cdot, \mathsf{sk})$ on M and forwards the reply to $\mathcal{A}$.

If $\mathcal{A}$ outputs $((m_1, \tau_1), \ldots, (m_{k+1}, \tau_{k+1}))$ with $\tau_i = (\sigma_i, R_i, T_i)$ after k successful queries to $(\cdot, \mathcal{S}(\mathsf{sk}))$ such that $m_i \neq m_j \quad \forall i, j \in [k+1], i \neq j$ and $\mathsf{Verify}(m_i, \tau_i, \mathsf{pk}) = 1 \ \forall\ i \in [k+1]$ then $\mathcal{B}$ aborts if for some $i \neq j \in [k+1]$: $m_iP + T_i = m_jP + T_j$ (we have a Type-2 forgery).

Otherwise, we have $(m_iP + T_i, P) \neq (m_jP + T_j, P)$ for all $i, j \in [k+1], i \neq j$. $\mathcal{A}$ has made k signing queries, but $((m_iP + T_i, P), \sigma_i)_{i\in[k+1]}$ are $k+1$ valid SPS-EQ message-signature pairs for *distinct* classes. Consequently, there must exist $i^* \in [k+1]$ such that the message-signature pair $((m_{i^*}P + T_{i^*}, P), \sigma_{i^*})$ represents a class that was not queried to $\mathcal{C}$'s signing oracle. Hence, one of these $k+1$ message-signature pairs enables $\mathcal{B}$ to break the EUF-CMA security of the SPS-EQ scheme. Due to the blindness, however, $\mathcal{B}$ cannot link the pairs to the messages $M_i = (s_i(m_iP + r_iQ), s_iP)$ which $\mathcal{A}$ has queried to the $(\cdot, \mathcal{S}(\mathsf{sk}))$ oracle. Therefore $\mathcal{B}$ guesses an index $i^* \in [k+1]$ and outputs $((m_{i^*}P + T_{i^*}, P), \sigma_{i^*})$ as a forgery to $\mathcal{C}$. If $\mathcal{A}$ wins the unforgeability game then $\mathcal{B}$ breaks the EUF-CMA security of $\mathsf{SPS\text{-}EQ}$ incurring a polynomial loss of $1/(k+1)$.

Type 2: $\mathcal{B}$ obtains an instance $(\mathsf{BG}, Q = qP, \hat{Q} = q\hat{P})$ of the co-DHI problem in $\mathbb{G}_1$, and its goal is to compute $q^{-1}P$. It computes $(\mathsf{sk}, \mathsf{pk}_{\mathcal{R}}) \xleftarrow{R} \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^\ell)$ for $\ell = 2$, and runs $\mathcal{A}$ on $\mathsf{pk} \leftarrow (\mathsf{pk}_{\mathcal{R}}, Q, \hat{Q})$ simulating its $(\cdot, \mathcal{S}(\mathsf{sk}))$ oracle as in the real game using sk.

If $\mathcal{A}$ outputs $(m_i, \tau_i = (\sigma_i, R_i, T_i))_{i\in[k+1]}$ after k successful oracle queries such that $m_i \neq m_j$ for all $1 \leq i < j \leq k+1$ and $\mathsf{Verify}(m_i, \tau_i, \mathsf{pk}) = 1$ for all $i \in [k+1]$, then $\mathcal{B}$ aborts if $m_iP + T_i \neq m_jP + T_j$ for all $i, j \in [k+1]$ (we have a Type-1 forgery).

Otherwise, let $i, j \in [k+1]$ be such that $m_iP + T_i = m_jP + T_j$ $(*)$. From the second equation in Verify, since $\hat{Q} = q\hat{P}$, we get $T_i = qR_i$ and $T_j = qR_j$. Together with $(*)$ we have $m_iP + qR_i = m_jP + qR_j$, that is $(m_i - m_j)P = q(R_j - R_i)$,

and since $m_i \neq m_j$: $q^{-1}P = (m_i - m_j)^{-1}(R_j - R_i)$. The latter, which $\mathcal{B}$ can efficiently compute, is, thus, a solution to the co-DHI problem in $\mathbb{G}_1$. □

Blindness

For the honest-signer model, blindness follows from the DDH assumption and perfect adaptation of signatures (cf. Definition 3.7) of the underlying SPS-EQ scheme $\mathsf{SPS\text{-}EQ}$. Let $Q \leftarrow qP$ and let q be part of the signing key, and let (P, rP, sP, tP) be a DDH instance. In the blindness game, we compute M as $(m \cdot sP + q \cdot tP, sP)$. When the adversary returns a signature on M, we must adapt it to the unblinded message—which we are unable to do as we do not know the blinding factor s. By perfect adaptation, however, an adapted signature is distributed as a fresh signature on the unblinded message, so, knowing the secret key, we can compute a signature σ on $(m \cdot P + q \cdot rP, P)$ and return the blind signature $(\sigma, rP, q \cdot rP)$. If the DDH instance was *real*, i.e., $t = s \cdot r$, then we perfectly simulated the game; if t was random then the adversary's view during issuing was independent of m.

For blindness in the malicious-signer model, we have to deal with two obstacles. (1) We do not have access to the adversarially generated signing key, meaning we cannot recompute the signature on the unblinded message. (2) The adversarially generated public-key values $Q, \hat{Q}$ do not allow us to embed a DDH instance for blinding and unblinding.

We overcome (1) by using the adversary $\mathcal{A}$ itself as a signing oracle by rewinding it. We first run $\mathcal{A}$ to obtain a signature on $(s'(mP+rQ), s'P)$, which, knowing s', we can transform into a signature on $(mP + rQ, P)$. We then rewind $\mathcal{A}$ to the point after outputting its public key and run it again, this time embedding our challenge. In the second run we cannot transform the received signature, instead we use the signature from the first run, which is distributed identically, due to perfect adaptation under malicious keys (Definition 3.8) of the SPS-EQ scheme.

To deal with the second obstacle, we use an interactive variant of the DDH assumption: Instead of being given BG, rP, sP and having to distinguish rsP from random, the adversary, for some Q of its choice (after being given BG), is given rP, rQ, sP and must distinguish rsQ from random.

Assumption 4.11. *Let* BGGen *be a bilinear-group generator that outputs* $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$. *We assume that for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l|rl} \mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(1^\kappa),\ b \xleftarrow{R} \{0,1\}, & & \\ (\mathsf{st}, Q, \hat{Q}) \xleftarrow{R} \mathcal{A}(\mathsf{BG}),\ r, s, t \xleftarrow{R} \mathbb{Z}_p, & : & e(Q, \hat{P}) = e(P, \hat{Q}) \\ b^* \xleftarrow{R} \mathcal{A}(\mathsf{st}, rP, rQ, sP, & & \wedge\ b^* = b \\ \qquad (b \cdot rs + (1-b) \cdot t)Q) & & \end{array}\right] - \frac{1}{2} \leq \epsilon(\kappa).$$

Proposition 4.12. *Assumption 4.11 holds in generic groups and reaches the optimal, quadratic simulation-error bound.*

The proof can be found in Appendix A.2.

Theorem 4.13. *If the underlying SPS-EQ scheme* SPS-EQ *perfectly adapts signatures under malicious keys and Assumption 4.11 holds, then Scheme 4 is blind.*

In the proof of blindness of our blind-signature scheme, we will use the following implication of Definition 3.8:

Corollary 4.14. *Let* SPS-EQ *be an SPS-EQ scheme on* $(\mathbb{G}_i^*)^\ell$ *that satisfies Definition 3.8. If for a tuple* $(\mathsf{pk}, M, s_0, s_1, \sigma_0, \sigma_1)$ *we have*

$$\mathsf{Verify}_{\mathcal{R}}(s_0 M, \sigma_0, \mathsf{pk}) = 1 \quad \textit{and} \quad \mathsf{Verify}_{\mathcal{R}}(s_1 M, \sigma_1, \mathsf{pk}) = 1$$

then $\mathsf{ChgRep}_{\mathcal{R}}(s_0 M, \sigma_0, \frac{1}{s_0}, \mathsf{pk})$ *and* $\mathsf{ChgRep}_{\mathcal{R}}(s_1 M, \sigma_1, \frac{1}{s_1}, \mathsf{pk})$ *are identically distributed.*

Proof. The statement follows, since for $b = 0, 1$ the tuple $(\mathsf{pk}, s_b M, \sigma_b, 1/s_b)$ satisfies (3.1) (in Definition 3.8), and for $(M, \sigma_b) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(s_b M, \sigma_b, 1/s_b, \mathsf{pk})$, by Definition 3.8 σ_b is random conditioned on $\mathsf{Verify}_{\mathcal{R}}(M, \sigma_b, \mathsf{pk}) = 1$. Thus σ_0 and σ_1 are identically distributed. □

Proof. Let $\mathbf{Exp}^{\text{blind}}$ be the blindness game (with adversarially/maliciously generated public keys) defined in Definition 4.5. Consider $\mathbf{Exp}^{\text{blind}}_{\mathcal{A},\mathsf{BS}}$ with BS being Scheme 4 and any PPT adversary $\mathcal{A}$, which we assume w.l.o.g. makes both calls to its $(\mathcal{U}(m_b, \mathsf{pk}), \cdot)$ oracle. Written out, we have:

$\mathbf{Exp}^{\text{blind}}_{\mathcal{A},\mathsf{BS}}$:
$b \xleftarrow{R} \{0,1\}$
$(\mathsf{st}_{\mathcal{A}}, (\mathsf{pk}_{\mathcal{R}}, Q, \hat{Q}), m_0, m_1) \xleftarrow{R} \mathcal{A}(1^\kappa)$
$\mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$
If $Q = 0_{\mathbb{G}_1}$ or $e(Q, \hat{P}) \neq e(P, \hat{Q})$ then $M_0, M_1 \leftarrow \perp$
Else
  $r_0, s_0 \xleftarrow{R} \mathbb{Z}_p$; $r_1, s_1 \xleftarrow{R} \mathbb{Z}_p$
  $M_0 \leftarrow (s_0(m_0 P + r_0 Q), s_0 P)$; $M_1 \leftarrow (s_1(m_1 P + r_1 Q), s_1 P)$
$(\mathsf{st}_{\mathcal{A}}, \pi_b) \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, M_b)$; $(\mathsf{st}_{\mathcal{A}}, \pi_{1-b}) \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, M_{1-b})$
If $(M_0, M_1) = (\perp, \perp)$ or $\mathsf{Verify}_{\mathcal{R}}(M_0, \pi_0, \mathsf{pk}) = 0$ or $\mathsf{Verify}_{\mathcal{R}}(M_1, \pi_1, \mathsf{pk}) = 0$
  then $b^* \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, \perp, \perp)$
Else
  $(N_0, \sigma_0) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M_0, \pi_0, 1/s_0, \mathsf{pk})$
  $(N_1, \sigma_1) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M_1, \pi_1, 1/s_1, \mathsf{pk})$
  $b^* \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, (\sigma_0, r_0 P, r_0 Q), (\sigma_1, r_1 P, r_1 Q))$
Return $(b^* = b)$

We have slightly modified the game, in that (for $i = 0, 1$) we allowed s_i to also take the value 0 and r_i to be such that $m_i P + r_i Q = 0_{\mathbb{G}_1}$. However, these events only happen with negligible probability.

We first argue that if $\mathcal{A}$ outputs an inconsistent public key or if π_0 or π_1 do not pass $\mathsf{Verify}_{\mathcal{R}}$ then the bit b is information-theoretically hidden from $\mathcal{A}$. This

is because if one of the above is the case then in the second phase $\mathcal{A}$ receives $(\bot, \bot)$, and r_0, r_1 information-theoretically hide m_0, m_1, and thus the bit b is also information-theoretically hidden, meaning $\Pr[\mathbf{Exp}^{\text{blind}}_{\mathcal{A},\mathsf{BS}} = 1] = 1/2$.

We can now assume w.l.o.g. that $\mathcal{A}$ outputs a valid pk and π_0 and π_1 verify: If $\mathcal{A}$ was not like this, we could construct a well-behaving adversary $\mathcal{A}'$ from $\mathcal{A}$: $\mathcal{A}'$ simulates $\mathcal{A}$ and whenever $\mathcal{A}$ misbehaves (which $\mathcal{A}'$ can efficiently detect), it aborts the simulation and outputs a random bit. By the above, $\mathcal{A}'$ wins with the same probability as $\mathcal{A}$. With this assumption on $\mathcal{A}$ the experiment simplifies thus to:

$\mathbf{Exp}^{\text{blind-non-}\bot}_{\mathcal{A},\mathsf{BS}}$:
$(\mathsf{st}_{\mathcal{A}}, (\mathsf{pk}_{\mathcal{R}}, Q, \hat{Q}), m_0, m_1) \xleftarrow{R} \mathcal{A}(1^\kappa)$
$\mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$
$r_0, r_1 \xleftarrow{R} \mathbb{Z}_p \qquad (*)$
$s_0, s_1 \xleftarrow{R} \mathbb{Z}_p \; ; \; b \xleftarrow{R} \{0,1\}$
$M_0 \leftarrow (s_0(m_0 P + r_0 Q), s_0 P)$
$M_1 \leftarrow (s_1(m_1 P + r_1 Q), s_1 P)$
$(\mathsf{st}_{\mathcal{A}}, \pi_b) \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, M_b)$
$(\mathsf{st}_{\mathcal{A}}, \pi_{1-b}) \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, M_{1-b})$
$(N_0, \sigma_0) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M_0, \pi_0, \frac{1}{s_0}, \mathsf{pk})$
$(N_1, \sigma_1) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M_1, \pi_1, \frac{1}{s_1}, \mathsf{pk})$
$b^* \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, (\sigma_0, r_0 P, r_0 Q), (\sigma_1, r_1 P, r_1 Q))$
Return $(b^* = b)$

Execution 1. Now we do the following: We run $\mathbf{Exp}^{\text{blind-non-}\bot}$ with $\mathcal{A}$, in particular choosing $r_0, r_1, s_0^{(1)}, s_1^{(1)}$ and $b^{(1)}$, constructing

$$M_0^{(1)} \leftarrow s_0^{(1)}\big((m_0 P + r_0 Q),\, P\big), \qquad M_1^{(1)} \leftarrow s_1^{(1)}\big((m_1 P + r_1 Q),\, P\big)$$

and running $\mathcal{A}$ on $M_{b^{(1)}}^{(1)}$ and then on $M_{1-b^{(1)}}^{(1)}$ to obtain signatures $\pi_0^{(1)}, \pi_1^{(1)}$. Then we *rewind* the experiment to the point $(*)$ and run it again. We choose independent uniform random $s_0^{(2)}, s_1^{(2)} \xleftarrow{R} \mathbb{Z}_p$, $b^{(2)} \xleftarrow{R} \{0,1\}$ (but use the same r_0, r_1 as in the first run), set

$$M_0^{(2)} \leftarrow s_0^{(2)}\big((m_0 P + r_0 Q),\, P\big), \qquad M_1^{(2)} \leftarrow s_1^{(2)}\big((m_1 P + r_1 Q),\, P\big),$$

run $\mathcal{A}$ on $M_{b^{(2)}}^{(2)}$ and then on $M_{1-b^{(2)}}^{(2)}$ to obtain signatures $\pi_0^{(2)}, \pi_1^{(2)}$, and finish the experiment: For $i = 0, 1$ we compute

$$(N_i^{(2)}, \sigma_i^{(2)}) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}\big(M_i^{(2)}, \pi_i^{(2)}, \tfrac{1}{s_i^{(2)}}, \mathsf{pk}\big), \quad \text{run } \; b^* \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, (\sigma_0^{(2)}, r_0 P, r_0 Q), (\sigma_1^{(2)}, r_1 P, r_1 Q)), \quad (4.1)$$

and return $(b^* = b^{(2)})$. As the second run simply constitutes an independent run of $\mathcal{A}$ we have that the probability of returning 1 is precisely $\Pr[\mathbf{Exp}^{\text{blind-non-}\bot}_{\mathcal{A},\mathsf{BS}} = 1] = \Pr[\mathbf{Exp}^{\text{blind}}_{\mathcal{A},\mathsf{BS}} = 1]$ (by our assumption on $\mathcal{A}$).

Execution 2. We now introduce a modification. We proceed as in Execution 1, but instead of (4.1), we compute for $i = 0, 1$:

$$(N_i^{(1)}, \sigma_i^{(1)}) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}\big(M_i^{(1)}, \pi_i^{(1)}, \tfrac{1}{s_i^{(1)}}, \mathsf{pk}\big),$$
$$\text{run } \; b^* \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, (\sigma_0^{(1)}, r_0 P, r_0 Q), (\sigma_1^{(1)}, r_1 P, r_1 Q)), \quad (4.2)$$

and return $(b^* = b^{(2)})$. That is, we use the signatures $\pi_0^{(1)}, \pi_1^{(1)}$ from the *first* run, adapt them to signatures on $N_i^{(1)} = (m_i P + r_i Q, P) = N_i^{(2)}$ and give them to $\mathcal{A}$ as part of our blind signatures. We now argue that the winning probability of the adversary does not change. For $i = 0, 1$ we have the following. Since by assumption we have

$$\mathsf{Verify}_{\mathcal{R}}\big(s_i^{(1)} \cdot (m_i P + r_i Q, P), \pi_i^{(1)}, \mathsf{pk}\big) = 1 \quad \text{and}$$
$$\mathsf{Verify}_{\mathcal{R}}\big(s_i^{(2)} \cdot (m_i P + r_i Q, P), \pi_i^{(2)}, \mathsf{pk}\big) = 1,$$

the tuple $\big(\mathsf{pk}, (m_i P + r_i Q, P), s_i^{(1)}, s_i^{(2)}, \pi_i^{(1)}, \pi_i^{(2)}\big)$ satisfies the premise of Corollary 4.14 and hence the outputs $\sigma_i^{(1)}$ and $\sigma_i^{(2)}$ of $\mathsf{ChgRep}_{\mathcal{R}}(M_i^{(1)}, \pi_i^{(1)}, {}^1\!/_{s_i^{(1)}}, \mathsf{pk})$ and $\mathsf{ChgRep}_{\mathcal{R}}(M_i^{(2)}, \pi_i^{(2)}, {}^1\!/_{s_i^{(2)}}, \mathsf{pk})$, respectively, are identically distributed. So, Executions 1 and 2 are identically distributed and the probability that after Execution 2 we have $(b^* = b^{(2)})$ is $\Pr[\mathbf{Exp}^{\text{blind}}_{\mathcal{A},\,\mathsf{BS}} = 1]$.

Let us write down Execution 2:

$(\mathsf{st}_{\mathcal{A}}, (\mathsf{pk}_{\mathcal{R}}, Q, \hat{Q}), m_0, m_1) \xleftarrow{R} \mathcal{A}(1^\kappa)$
$\mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$
$r_0, r_1 \xleftarrow{R} \mathbb{Z}_p \qquad (*)$

$s_0^{(1)}, s_1^{(1)} \xleftarrow{R} \mathbb{Z}_p \; ; \; b^{(1)} \xleftarrow{R} \{0,1\}$	$s_0^{(2)}, s_1^{(2)} \xleftarrow{R} \mathbb{Z}_p \; ; \; b^{(2)} \xleftarrow{R} \{0,1\}$
$M_0^{(1)} \leftarrow s_0^{(1)}\big((m_0 P + r_0 Q),\, P\big)$	$M_0^{(2)} \leftarrow s_0^{(2)}\big((m_0 P + r_0 Q),\, P\big)$ $\Big\Vert$
$M_1^{(1)} \leftarrow s_1^{(1)}\big((m_1 P + r_1 Q),\, P\big)$	$M_1^{(2)} \leftarrow s_1^{(2)}\big((m_1 P + r_1 Q),\, P\big)$ $\Big\Vert$
$(\mathsf{st}'_{\mathcal{A}}, \pi_{b^{(1)}}^{(1)}) \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, M_{b^{(1)}}^{(1)})$	$(\mathsf{st}_{\mathcal{A}}, \pi_{b^{(2)}}^{(2)}) \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, M_{b^{(2)}}^{(2)})$
$(\mathsf{st}'_{\mathcal{A}}, \pi_{1-b^{(1)}}^{(1)}) \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, M_{1-b^{(1)}}^{(1)})$	$(\mathsf{st}_{\mathcal{A}}, \pi_{1-b^{(2)}}^{(2)}) \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, M_{1-b^{(2)}}^{(2)})$

$(N_0, \sigma_0) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M_0^{(1)}, \pi_0^{(1)}, \frac{1}{s_0^{(1)}}, \mathsf{pk})$
$(N_1, \sigma_1) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M_1^{(1)}, \pi_1^{(1)}, \frac{1}{s_1^{(1)}}, \mathsf{pk})$
$b^* \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, (\sigma_0, r_0 P, r_0 Q), (\sigma_1, r_1 P, r_1 Q))$
Return $(b^* = b^{(2)})$

Execution 3. We define another variant, where in Execution 2 we replace the two lines marked with $\Vert$ by

$$t_0 \xleftarrow{R} \mathbb{Z}_p \,;\, M_0^{(2)} \leftarrow \big(s_0^{(2)} m_0 P + t_0 Q,\; s_0^{(2)} P\big) \;\Big\Vert$$
$$M_1^{(2)} \leftarrow \big(s_1^{(2)} m_1 P + s_1^{(2)} r_1 Q,\; s_1^{(2)} P\big) \;\Big\Vert$$

that is, in the definition of $M_0^{(2)}$ we replaced the value $s_0^{(2)} r_0$ with a random element t_0.

Execution 4. Our final execution also replaces $s_1^{(2)} r_1$ in the definition of $M_1^{(2)}$ with a random element t_1. That is, it is defined as Execution 2 above, except with the lines marked with $\|$ replaced by the following:

$$\begin{array}{l} t_0 \xleftarrow{R} \mathbb{Z}_p \,;\, M_0^{(2)} \leftarrow \big(s_0^{(2)} m_0 P + t_0 Q,\, s_0^{(2)} P\big) \\ t_1 \xleftarrow{R} \mathbb{Z}_p \,;\, M_1^{(2)} \leftarrow \big(s_1^{(2)} m_1 P + t_1 Q,\, s_1^{(2)} P\big) \end{array} \Bigg\|$$

Claim 4.15. *If Assumption 4.11 holds then Executions 2 and 3 are indistinguishable; likewise, Executions 3 and 4 are indistinguishable.*

Proof. Assume that there exists an adversary $\mathcal{A}$, for whom the probability that $(b^* = b^{(2)})$ is noticeably different in Executions 2 and 3. Then we construct an adversary $\mathcal{B}$ against the Assumption 4.11 as follows:

On input 1^κ, $\mathcal{B}$ runs $(\mathsf{st}_\mathcal{A}, (\mathsf{pk}_\mathcal{R}, Q, \hat{Q}), m_0, m_1) \xleftarrow{R} \mathcal{A}(1^\kappa)$ and outputs

$$(\mathsf{st}_\mathcal{B} \leftarrow (\mathsf{st}_\mathcal{A}, \mathsf{pk}_\mathcal{R}, m_0, m_1), Q, \hat{Q}) \;;$$

$\mathcal{B}$ then receives a challenge (rP, rQ, sP, tQ) and needs to decide whether $t = rs$. $\mathcal{B}$ simulates Execution 2 with $\mathcal{A}$, except that it implicitly sets $r_0 \leftarrow r$ and $s_0^{(2)} \leftarrow s$ as well as $s_0^{(2)} r_0 \leftarrow t$ from the assumption. $\mathcal{B}$'s output is $(b^* = b^{(2)})$. If $t = rs$ then $\mathcal{B}$ simulated Execution 2, whereas if t is uniformly random, it simulated Execution 3. In particular, after receiving the challenge, $\mathcal{B}$ runs as follows (which shows that the simulation can be done using the challenge, which we underline):

$\underline{\mathcal{B}\big(\mathsf{st}_\mathcal{B} = (\mathsf{st}_\mathcal{A}, \mathsf{pk}_\mathcal{R}, m_0, m_1), rP, rQ, sP, tQ\big):}$

$r_1 \xleftarrow{R} \mathbb{Z}_p \qquad (*)$
$s_0^{(1)}, s_1^{(1)} \xleftarrow{R} \mathbb{Z}_p \;;\; b^{(1)} \xleftarrow{R} \{0,1\}$
$M_0^{(1)} \leftarrow s_0^{(1)}\big((m_0 P + \underline{(rQ)}),\, P\big)$
$M_1^{(1)} \leftarrow s_1^{(1)}\big((m_1 P + r_1 Q),\, P\big)$
$(\mathsf{st}'_\mathcal{A}, \pi_{b^{(1)}}^{(1)}) \xleftarrow{R} \mathcal{A}(\mathsf{st}_\mathcal{A}, M_{b^{(1)}}^{(1)})$
$(\mathsf{st}'_\mathcal{A}, \pi_{1-b^{(1)}}^{(1)}) \xleftarrow{R} \mathcal{A}(\mathsf{st}_\mathcal{A}, M_{1-b^{(1)}}^{(1)})$
REWIND to $(*)$
  $s_1^{(2)} \xleftarrow{R} \mathbb{Z}_p \;;\; b^{(2)} \xleftarrow{R} \{0,1\}$
  $M_0^{(2)} \leftarrow \big(m_0 \underline{(sP)} + \underline{(tQ)},\, \underline{(sP)}\big)$
  $M_1^{(2)} \leftarrow s_1^{(2)}\big(m_1 P + r_1 Q,\, P\big)$
  $(\mathsf{st}_\mathcal{A}, \pi_{b^{(2)}}^{(2)}) \xleftarrow{R} \mathcal{A}(\mathsf{st}_\mathcal{A}, M_{b^{(2)}}^{(2)})$
  $(\mathsf{st}_\mathcal{A}, \pi_{1-b^{(2)}}^{(2)}) \xleftarrow{R} \mathcal{A}(\mathsf{st}_\mathcal{A}, M_{1-b^{(2)}}^{(2)})$
$(N_0, \sigma_0) \xleftarrow{R} \mathsf{ChgRep}_\mathcal{R}(M_0^{(1)}, \pi_0^{(1)}, \frac{1}{s_0^{(1)}}, \mathsf{pk})$

$(N_1, \sigma_1) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M_1^{(1)}, \pi_1^{(1)}, \frac{1}{s_1^{(1)}}, k)$
$b^* \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, (\sigma_0, \underline{(rP)}, \underline{(rQ)}), (\sigma_1, r_1 P, r_1 Q))$
Return $(b^* = b^{(2)})$

We have thus that the probability that $\mathcal{B}$ outputs 1 when given a DDH instance is the probability that Execution 2 outputs 1; and the probability that $\mathcal{B}$ outputs 1 when given a random instance is the probability that Execution 3 outputs 1. Thus, if $\mathcal{A}$ behaved differently in Executions 2 and 3 then $\mathcal{B}$ would break the assumption.

Analogously we can construct an adversary $\mathcal{B}$ which breaks the assumption given an adversary $\mathcal{A}$ that distinguishes Executions 3 and 4. □

Finally, let us consider Execution 4; that is:

$(\mathsf{st}_{\mathcal{A}}, (\mathsf{pk}_{\mathcal{R}}, Q, \hat{Q}), m_0, m_1) \xleftarrow{R} \mathcal{A}(1^\kappa)$
$\mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$
$r_0, r_1 \xleftarrow{R} \mathbb{Z}_p$

$s_0^{(1)}, s_1^{(1)} \xleftarrow{R} \mathbb{Z}_p$	$s_0^{(2)}, s_1^{(2)}, t_0, t_1 \xleftarrow{R} \mathbb{Z}_p$
$b^{(1)} \xleftarrow{R} \{0,1\}$	$b^{(2)} \xleftarrow{R} \{0,1\}$
$M_0^{(1)} \leftarrow \left(s_0^{(1)}(m_0 P + r_0 Q),\ s_0^{(1)} P\right)$	$M_0^{(2)} \leftarrow \left(s_0^{(2)} m_0 P + t_0 Q,\ s_0^{(2)} P\right)$
$M_1^{(1)} \leftarrow \left(s_1^{(1)}(m_1 P + r_1 Q),\ s_1^{(1)} P\right)$	$M_1^{(2)} \leftarrow \left(s_1^{(2)} m_1 P + t_1 Q,\ s_1^{(2)} P\right)$
$(\mathsf{st}'_{\mathcal{A}}, \pi_{b^{(1)}}^{(1)}) \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, M_{b^{(1)}}^{(1)})$	$(\mathsf{st}_{\mathcal{A}}, \pi_{b^{(2)}}^{(2)}) \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, M_{b^{(2)}}^{(2)})$
$(\mathsf{st}'_{\mathcal{A}}, \pi_{1-b^{(1)}}^{(1)}) \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, M_{1-b^{(1)}}^{(1)})$	$(\mathsf{st}_{\mathcal{A}}, \pi_{1-b^{(2)}}^{(2)}) \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, M_{1-b^{(2)}}^{(2)})$

$(N_0, \sigma_0) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M_0^{(1)}, \pi_0^{(1)}, \frac{1}{s_0^{(1)}}, \mathsf{pk})$
$(N_1, \sigma_1) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M_1^{(1)}, \pi_1^{(1)}, \frac{1}{s_1^{(1)}}, \mathsf{pk})$
$b^* \xleftarrow{R} \mathcal{A}(\mathsf{st}_{\mathcal{A}}, (\sigma_0, r_0 P, r_0 Q), (\sigma_1, r_1 P, r_1 Q))$
Return $(b^* = b^{(2)})$

We now see that for $i = 0, 1$, since $s_i^{(2)}$ and t_i are uniformly random and used nowhere other than in the definition of $M_i^{(2)}$, the latter is a uniform random element from $\mathbb{G}_1 \times \mathbb{G}_1$. Since $b^{(2)}$ is only used to determine the order in which $M_0^{(2)}$ and $M_1^{(2)}$ (which are both random elements) are sent to $\mathcal{A}$, the bit $b^{(2)}$ is information-theoretically hidden. We thus have that the probability that $(b^* = b^{(2)})$ in Execution 4 is exactly $1/2$.

Overall, we have that $\Pr[\mathbf{Exp}^{\text{blind}}_{\mathcal{A},\mathsf{BS}} = 1]$ can only be negligibly different from $1/2$, which proves blindness. □

4.2.2 Discussion

Basing Our Scheme on Non-Interactive Assumptions

Fischlin and Schröder [FS10] show that the unforgeability of a blind-signature scheme cannot be based on non-interactive hardness assumptions if (1) the scheme has 3 moves or less, (2) its blindness holds statistically and (3) from a transcript one can efficiently decide whether the interaction yielded a valid blind signature. Our scheme satisfies (1) and (3), whereas blindness only holds computationally.

They extend their result in [FS10] to computationally blind schemes that meet the following conditions: (4) One can efficiently check whether a public key has a matching secret key; this is the case in our setting because of group-membership tests and pairings. (5) Blindness needs to hold relative to a forgery oracle. As written in [FS10], this does, e.g., not hold for Abe's scheme [Abe01], where unforgeability is based on the DL problem and blindness on the DDH problem.

This is the case in our construction too (as one can forge signatures by solving discrete logarithms), hence the impossibility result does not apply to our scheme. Our blind-signature construction is black-box from any SPS-EQ with perfect adaptation under malicious keys (Definition 3.8). However, the only known such scheme is the one from [FHS14] (given in Scheme 2), which is EUF-CMA secure in the GGM, that is, it is based on an interactive assumption. Plugging this scheme into Scheme 4 yields a round-optimal blind-signature scheme with unforgeability under this interactive assumption and co-DHI, and blindness (under adversarially chosen keys) under Assumption 4.11, which is also interactive.

To construct a scheme under non-interactive assumptions, we would thus have to base blindness on a non-interactive assumption; and find an SPS-EQ scheme satisfying Definition 3.8 whose unforgeability is proven under a non-interactive assumption. We refer the reader to Section 3.6 for a detailed discussion of SPS-EQ unforgeability and further directions.

Efficiency of the Construction

When instantiating our blind-signature construction with the SPS-EQ given in Scheme 2, which we showed optimal, this yields a public-key size of $1\ \mathbb{G}_1 + 3\ \mathbb{G}_2$, a communication complexity of $4\ \mathbb{G}_1 + 1\ \mathbb{G}_2$ and a signature size of $4\ \mathbb{G}_1 + 1\ \mathbb{G}_2$ elements. For an 80-bit security setting, a blind signature has thus 120 Bytes.

The most efficient scheme from standard assumptions is based on DLIN [GG14]. Ignoring the increase of the security parameter due to complexity leveraging, their scheme has a public key size of 43 $\mathbb{G}_1$ elements, communication complexity $18 \log_2 q + 41\ \mathbb{G}_1$ elements (where, e.g., we have $\log_2 q = 155$ when assuming that the adversary runs in $\leq 2^{80}$ steps) and a signature size of 183 $\mathbb{G}_1$ elements.

4.3 Extension to Partially Blind Signatures

We now show how to construct a round-optimal PBS scheme PBS secure in the standard model from an SPS-EQ scheme SPS-EQ by modifying Scheme 4 as follows.

To include common information $\gamma \in \mathbb{Z}_p^*$, SPS-EQ is set up for $\ell = 3$. On input $M \leftarrow (s(mP+rQ), sP)$, $\mathcal{S}$ returns a signature for $M \leftarrow (s(mP+rQ), \gamma \cdot sP, sP)$ and $\mathcal{U}^{(2)}$ additionally checks correctness of the included γ and returns $\perp$ if this is not the case. Otherwise, it runs $((mP + rQ, \gamma P, P), \sigma) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M, \pi, \frac{1}{s}, \mathsf{pk})$ and outputs signature $\tau \leftarrow (\sigma, rP, rQ)$ for message m and common information γ.

For this construction we obtain the following theorems, whose proofs are analogous to those for Scheme 4 and thus omitted.

Theorem 4.16. *If* SPS-EQ *is EUF-CMA secure and the co-DHI assumption holds in* $\mathbb{G}_1$*, then the resulting PBS scheme is unforgeable.*

Theorem 4.17. *If* SPS-EQ *has perfect adaptation of signatures under malicious keys and Assumption 4.11 holds, then the resulting PBS scheme is partially blind.*

4.4 Blind Signatures on Message Vectors

In order to build one-show credentials later on (cf. Section 7.1), we introduce the following generalization of our blind-signature scheme from Section 4.2. Instead of signing single messages, it allows to sign message vectors in a way that enables an efficient coupling with PoKs.

In particular, our construction BSV of round-optimal blind signatures on message vectors $\vec{m} \in \mathbb{Z}_p^n$ simply replaces the Pedersen commitment $mP + rQ$ in Scheme 4 with a generalized Pedersen commitment $\sum_{i\in[n]} m_i P_i + rQ$. Thus, KeyGen, on input $(1^\kappa, 1^n)$, additionally outputs generators $(P_i)_{i\in[n]} \in \mathbb{G}_1^n$ and $\mathsf{Verify}(\vec{m}, (\sigma, R, T), \mathsf{pk})$ checks

$$\mathsf{Verify}_{\mathcal{R}}((\textstyle\sum_{i\in[n]} m_i P_i + T, P), \sigma, \mathsf{pk}_{\mathcal{R}}) = 1 \quad \text{and} \quad e(T, \hat{P}) = e(R, \hat{Q}).$$

The construction is presented in Scheme 5.

The predicate $\mathsf{Check}(\mathsf{pp})$ checks for valid commitment parameters: For a generalized Pedersen commitment in $\mathbb{G}_1$ of a Type-3 bilinear group BG with tweaked opening, we have $\mathsf{pp} = ((P_i)_{i\in[n]}, Q, \hat{Q})$. It returns 1 if the following holds

- $\mathsf{pp} \in (\mathbb{G}_1^*)^{n+1} \times \mathbb{G}_2^*$,
- all $\mathbb{G}_1$-elements are pairwise distinct, and
- $e(Q, \hat{P}) = e(P, \hat{Q})$,

and 0 otherwise.

Again, let SPS-EQ be the underlying SPS-EQ scheme.

Scheme 5 A blind-signature scheme on message vectors from SPS-EQ.

$\mathsf{KeyGen}(1^\kappa, 1^n)$: Given a security parameter 1^κ and vector length n in unary, compute $\mathsf{BG} \leftarrow \mathsf{BGGen}_\mathcal{R}(1^\kappa)$, $(\mathsf{sk}, \mathsf{pk}_\mathcal{R}) \xleftarrow{R} \mathsf{KeyGen}_\mathcal{R}(\mathsf{BG}, 1^\ell)$ for $\ell = 2$, pick $q \xleftarrow{R} \mathbb{Z}_p^*$ and $(p_i)_{i\in[n]} \xleftarrow{R} (\mathbb{Z}_p^*)^n$, set $Q \leftarrow qP$, $\hat{Q} \leftarrow q\hat{P}$ and $(P_i)_{i\in[n]} \leftarrow (p_i P)_{i\in[n]}$ and output $(\mathsf{sk}, \mathsf{pk} = (\mathsf{pk}_\mathcal{R}, (P_i)_{i\in[n]}, Q, \hat{Q}))$.

$\mathcal{U}^{(1)}(\vec{m}, \mathsf{pk})$: Given a message vector $\vec{m} \in \mathbb{Z}_p^n$ and a public key $\mathsf{pk} = (\mathsf{pk}_\mathcal{R}, (P_i)_{i\in[n]}, Q, \hat{Q})$, compute $\mathsf{BG} \leftarrow \mathsf{BGGen}_\mathcal{R}(1^\kappa)$. If $\mathsf{Check}((P_i)_{i\in[n]}, Q, \hat{Q}) = 0$ then return $\perp$; else choose $s \xleftarrow{R} \mathbb{Z}_p^*$ and $r \xleftarrow{R} \mathbb{Z}_p$ such that $\sum_{i\in[n]} m_i P_i + rQ \neq 0_{\mathbb{G}_1}$ and output

$$M \leftarrow \big(s(\textstyle\sum_{i\in[n]} m_i P_i + rQ), sP\big) \qquad \mathsf{st} \leftarrow (\mathsf{BG}, \mathsf{pk}_\mathcal{R}, Q, M, r, s).$$

$\mathcal{S}(M, \mathsf{sk})$: Given $M \in (\mathbb{G}_1^*)^2$ and a secret key sk, output $\pi \xleftarrow{R} \mathsf{Sign}_\mathcal{R}(M, \mathsf{sk})$.

$\mathcal{U}^{(2)}(\mathsf{st}, \pi)$: Given state st and π, parse st as $(\mathsf{BG}, \mathsf{pk}_\mathcal{R}, Q, M, r, s)$. If $\mathsf{Verify}_\mathcal{R}(M, \pi, \mathsf{pk}_\mathcal{R}) = 0$, return $\perp$. Else, run $((\sum_{i\in[n]} m_i P_i + rQ, P), \sigma) \xleftarrow{R} \mathsf{ChgRep}_\mathcal{R}(M, \pi, \frac{1}{s}, \mathsf{pk}_\mathcal{R})$ and output $\tau \leftarrow (\sigma, rP, rQ)$.

$\mathsf{Verify}(\vec{m}, \tau, \mathsf{pk})$: Given a message vector $\vec{m} \in \mathbb{Z}_p^n$, a blind signature $\tau = (\sigma, R, T)$ and a public key $\mathsf{pk} = (\mathsf{pk}_\mathcal{R}, (P_i)_{i\in[n]}, Q, \hat{Q})$ with $\mathsf{Check}((P_i)_{i\in[n]}, Q, \hat{Q}) = 1$, output 1 if the following holds and 0 otherwise.

$$\mathsf{Verify}_\mathcal{R}\big((\textstyle\sum_{i\in[n]} m_i P_i + T, P), \sigma, \mathsf{pk}_\mathcal{R}\big) = 1 \qquad e(T, \hat{P}) = e(R, \hat{Q})$$

4.4.1 Security

It is straightforward to generalize blind-signature security models to blind signatures on message vectors. In these models, we can prove the following, where the correctness of the scheme, again, follows by inspection.

Theorem 4.18. *If* SPS-EQ *is EUF-CMA secure and the co-DHI assumption holds in* $\mathbb{G}_1$*, then Scheme 5 is unforgeable.*

Proof (Sketch). The proof is analogous to the unforgeability proof of Scheme 4, except that for Type-2 adversaries, the reduction obtains $q^{-1}P$ from the relation

$$(r_j^* - r_i^*)P = \frac{(\sum_{k\in[n]} m_{i,k}^* p_k - \sum_{k\in[n]} m_{j,k}^* p_k)}{q} P,$$

which is implied by the following:

$$\begin{aligned} M_i^* - M_j^* &= \big(\sum_{k\in[n]} m_{i,k}^* P_k - \sum_{k\in[n]} m_{j,k}^* P_k \big) + (r_i^* - r_j^*)Q \\ &= \big(\sum_{k\in[n]} m_{i,k}^* p_k - \sum_{k\in[n]} m_{j,k}^* p_k \big) P + (r_i^* - r_j^*)Q. \end{aligned}$$

□

Theorem 4.19. *If* SPS-EQ *has perfect adaptation of signatures under malicious keys and Assumption 4.11 holds, then Scheme 5 is blind.*

The proof is identical to the blindness proof of Scheme 4 and thus omitted.

5

Verifiably Encrypted Signatures: Security Revisited and a Construction via SPS-EQ

The protection provided by encryption is based on the fact that most people would rather eat liver than do mathematics.
— Bill Neugent

Optimistic fair exchange gives two negotiating parties a guarantee that they will finally receive what they agreed upon or that neither party will. A non-interactive solution to this problem is provided by VESs, which have been introduced by Boneh, Gentry, Lynn and Shacham in 2003 [BGLS03]. Less desirable, interactive protocols to this end have already been given, e.g., in [ASW98] in 1998.

In order to achieve this goal, VESs feature two kinds of signatures: encrypted, but still verifiable, and plain (i.e., standard) signatures. A common scenario is the following one. Let Bob and Camilla be two parties agreeing on a deal. In doing so, Bob sends an encrypted signature ω on a document (e.g., a transaction receipt) to Camilla in order to prove to Camilla his willingness to fulfill his side of the deal. Camilla can verify ω, but not use ω any further. Once Camilla has met her obligations, Bob sends the plain signature σ corresponding to ω to Camilla. In case of a dispute, there is a third party Alice, taking the role of an arbiter, who is able to extract σ from ω, if Bob denies doing so. In sum, *fairness* ensures that, at the end of the protocol, all participating parties will either obtain the expected items or that none of them will receive anything. *Optimistic* means that the third party Alice only gets involved if necessary.

In this chapter, we start by revisiting the security of VESs, point out some

shortcomings in the security models and show how to repair them.

Then, we present a new black-box standard-model construction of verifiably encrypted signatures from SPS-EQ. The construction follows the idea that each message is associated with a projective equivalence class and that encrypted and plain signatures are just signatures on different representatives of the same class—whose relation depends on the arbiter key: The arbiter secret key is used to scale between encrypted and plain signatures.

We further introduce a new property characterizing SPS-EQs, called perfect composition (Definition 5.15). We show that it is a necessary criterion for the underlying SPS-EQ in order to apply the transformation from Calderon et al. [CMSW14], which turns a VES into a PKE scheme. This means that perfectly composing SPS-EQ schemes imply PKE. This is not only interesting because it draws a connection between SPS-EQ and PKE, but, at the same time, also because it separates such SPS-EQs from OWFs (i.e., shows that such schemes cannot be built black-box from OWFs).

The results in this chapter are joint work with Max Rabkin and Dominique Schröder. The material in this chapter is based on [HRS15]. Before stating the main contribution, we recall the basic definitions in this context.

5.1 Verifiably Encrypted Signatures: Basic Definitions

We now give the formal models of verifiably encrypted signatures [BGLS03, CMSW14].

Definition 5.1 (Verifiably-encrypted signature (VES) scheme)**.** *A* VES scheme VES *consists of the following PPT algorithms:*

$\mathsf{AKeyGen}(1^\kappa)$: *A probabilistic algorithm that takes input a security parameter* 1^κ*. It returns an arbiter key pair* $(\mathsf{ask}, \mathsf{apk})$*.*

$\mathsf{KeyGen}(1^\kappa)$: *A probabilistic algorithm that takes input a security parameter* 1^κ*. It returns a signer key pair* $(\mathsf{sk}, \mathsf{pk})$ *(we assume that* pk *includes a description of the message space* $\mathcal{M}_{\mathsf{pk}}$*).*

$\mathsf{Sign}(m, \mathsf{sk})$: *A (probabilistic) algorithm that takes input a message* $m \in \mathcal{M}_{\mathsf{pk}}$ *and a signing key* sk*. It returns a signature* σ *under* sk *on* m*.*

$\mathsf{Verify}(m, \sigma, \mathsf{pk})$: *A deterministic algorithm that takes input a message* $m \in \mathcal{M}_{\mathsf{pk}}$*, a signature* σ *and a signer public key* pk*. It returns* 1 *or* 0 *indicating whether or not* σ *is a valid signature on* m *under* pk*.*

$\mathsf{VESign}(m, \mathsf{sk}, \mathsf{apk})$: *A (probabilistic) algorithm that takes input a message* $m \in \mathcal{M}_{\mathsf{pk}}$*, a signing key* sk *and an arbiter public key* apk*. It returns an encrypted signature* ω *under* sk *on message* m*.*

$\mathsf{VEVerify}(m, \omega, \mathsf{pk}, \mathsf{apk})$: *A deterministic algorithm that takes input a message* $m \in \mathcal{M}_{\mathsf{pk}}$*, an encrypted signature* ω*, a signer public key* pk *and an arbiter public key* apk*. It returns* 1 *or* 0 *indicating whether or not* ω *is a valid encrypted signature on* m *under* pk*.*

$\mathsf{Resolve}(m, \omega, \mathsf{ask}, \mathsf{pk})$: *A (probabilistic) algorithm that takes input a message* $m \in \mathcal{M}_{\mathsf{pk}}$*, an encrypted signature* ω*, an arbiter secret key* ask *and a signer public key* pk*. It returns a signature* σ *on* m *under* pk*, which is extracted from* ω*.*

We call a VES *secure* if it is *complete*, *unforgeable*, *opaque*, *extractable*, *abuse free* and *resolution independent*. We define these properties below.

Completeness demands that honestly computed VESs always verify and that the arbiter can always pull out a valid signature.

Definition 5.2 (Completeness)**.** *A VES scheme* VES *is* complete *if for all* $\kappa > 0$*, all choices of* $(\mathsf{ask}, \mathsf{apk}) \xleftarrow{R} \mathsf{AKeyGen}(1^\kappa)$*, all choices of* $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa)$ *and all* $m \in \mathcal{M}_{\mathsf{pk}}$ *it holds that:*

$$\Pr\left[\begin{array}{l} \omega \xleftarrow{R} \mathsf{VESign}(m, \mathsf{sk}, \mathsf{apk}), \\ \sigma \xleftarrow{R} \mathsf{Resolve}(m, \omega, \mathsf{ask}, \mathsf{pk}) \end{array} : \begin{array}{r} \mathsf{VEVerify}(m, \omega, \mathsf{pk}, \mathsf{apk}) = 1 \\ \wedge\ \mathsf{Verify}(m, \sigma, \mathsf{pk}) = 1 \end{array}\right] = 1.$$

Unforgeability says that it should be infeasible to produce a valid encrypted signature for an unknown secret key.

Definition 5.3 (Unforgeability)**.** *A VES scheme* VES *is* unforgeable *if for all PPT adversaries* $\mathcal{A}$ *having oracle access to* $\mathcal{O} := \{\mathsf{Sign}(\cdot, \mathsf{sk}), \mathsf{VESign}(\cdot, \mathsf{sk}, \mathsf{apk}), \mathsf{Resolve}(\cdot, \cdot, \mathsf{ask}, \mathsf{pk})\}$*, there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l} (\mathsf{ask}, \mathsf{apk}) \xleftarrow{R} \mathsf{AKeyGen}(1^\kappa), \\ (\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa), \\ (m^*, \omega^*) \xleftarrow{R} \mathcal{A}^{\mathcal{O}}(\mathsf{pk}, \mathsf{apk}) \end{array} : \begin{array}{r} \mathsf{VEVerify}(m^*, \omega^*, \mathsf{pk}, \mathsf{apk}) = 1 \\ \wedge\ m^* \notin Q \end{array}\right] \leq \epsilon(\kappa),$$

where Q *is the set of messages which were queried to the oracles.*

Opacity essentially requires that only the arbiter should be able to extract the underlying signature.

Definition 5.4 (Opacity)**.** *A VES scheme* VES *is* opaque *if for all PPT adversaries* $\mathcal{A}$ *having oracle access to* $\mathcal{O} := \{\mathsf{VESign}(\cdot, \mathsf{sk}, \mathsf{apk}), \mathsf{Resolve}(\cdot, \cdot, \mathsf{ask}, \mathsf{pk})\}$*, there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l} (\mathsf{ask}, \mathsf{apk}) \xleftarrow{R} \mathsf{AKeyGen}(1^\kappa), \\ (\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa), \\ (m^*, \sigma^*) \xleftarrow{R} \mathcal{A}^{\mathcal{O}}(\mathsf{pk}, \mathsf{apk}) \end{array} : \begin{array}{r} \mathsf{Verify}(m^*, \sigma^*, \mathsf{pk}) = 1 \\ \wedge\ m^* \notin Q \end{array}\right] \leq \epsilon(\kappa),$$

where Q *is the set of messages queried to the* $\mathsf{Resolve}$ *oracle.*

In addition to the above property, we need to ensure that it is indeed possible for the arbiter to extract the underlying signature:

Definition 5.5 (Extractability)**.** *A VES scheme* VES *is* extractable *if for all PPT adversaries* $\mathcal{A}$ *having oracle access to* $\mathcal{O} := \{\mathsf{Resolve}(\cdot,\cdot,\mathsf{ask},\cdot)\}$*, there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l}(\mathsf{ask},\mathsf{apk}) \xleftarrow{R} \mathsf{AKeyGen}(1^\kappa),\\ (\mathsf{pk}^*, m^*, \omega^*) \xleftarrow{R} \mathcal{A}^{\mathcal{O}}(\mathsf{apk}),\\ \sigma \xleftarrow{R} \mathsf{Resolve}(m^*, \omega^*, \mathsf{ask}, \mathsf{pk}^*)\end{array} : \begin{array}{r}\mathsf{VEVerify}(m^*, \omega^*, \mathsf{pk}^*, \mathsf{apk}) = 1\\ \wedge\ \mathsf{Verify}(m^*, \sigma, \mathsf{pk}^*) = 0\end{array}\right] \leq \epsilon(\kappa).$$

Abuse freeness guarantees that, even if the arbiter is colluding with the adversary, the adversary is still not able to forge a valid encrypted signature.

Definition 5.6 (Abuse freeness)**.** *A VES scheme* VES *is* abuse-free *if for all PPT adversaries* $\mathcal{A}$ *having oracle access to* $\mathcal{O} := \{\mathsf{VESign}(\cdot,\mathsf{sk},\mathsf{apk})\}$*, there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l}(\mathsf{ask},\mathsf{apk}) \xleftarrow{R} \mathsf{AKeyGen}(1^\kappa),\\ (\mathsf{sk},\mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa),\\ (m^*, \omega^*) \xleftarrow{R} \mathcal{A}^{\mathcal{O}}(\mathsf{pk},\mathsf{ask},\mathsf{apk})\end{array} : \begin{array}{r}\mathsf{VEVerify}(m^*, \omega^*, \mathsf{pk}, \mathsf{apk}) = 1\\ \wedge\ m^* \notin Q\end{array}\right] \leq \epsilon(\kappa),$$

where Q *is the set of messages queried to the* VESign *oracle.*

The two latter properties were introduced by Rückert and Schröder [RS09].

Calderon et al. [CMSW14] have identified an additional property called resolution independence. It demands that plain signatures and extracted signatures are identically distributed, which prevents discrimination between signatures arising from signers and arbiters. In Section 5.2, we will see that this property is crucial to the security of a VES.

Definition 5.7 (Resolution independence)**.** *A VES scheme* VES *is* resolution independent *if for all* $\kappa > 0$*, all choices of* $(\mathsf{ask},\mathsf{apk}) \xleftarrow{R} \mathsf{AKeyGen}(1^\kappa)$*, all choices of* $(\mathsf{sk},\mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa)$*, and all messages* $m \in \mathcal{M}_{\mathsf{pk}}$*, the outputs of* $\mathsf{Sign}(m,\mathsf{sk})$ *and* $\mathsf{Resolve}(m, \mathsf{VESign}(m,\mathsf{sk},\mathsf{apk}),\mathsf{ask},\mathsf{pk})$ *are distributed identically.*

In [CMSW14], the authors showed that VESs imply PKE, if they satisfy an even stronger property called *resolution duplication*. Informally, a VES is resolution duplicate if the signatures generated by the signer and the arbiter are indeed identical.

Definition 5.8 (Resolution duplication)**.** *A VES scheme* VES *is* resolution duplicate *if it is resolution independent and fulfills the following properties:*

Deterministic Resolution: *The algorithm* $\mathsf{Resolve}$ *is deterministic.*

Extraction: *There exists an additional PPT algorithm* $\mathsf{Extract}(\cdot,\cdot,\cdot)$*, such that for all* $\kappa > 0$*, all choices of* $(\mathsf{ask},\mathsf{apk}) \xleftarrow{R} \mathsf{AKeyGen}(1^\kappa)$*, all choices of* $(\mathsf{sk},\mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa)$*, all* $m \in \mathcal{M}_{\mathsf{pk}}$ *and random tapes* $r \in \{0,1\}^*$*, it is the case that*

$$\mathsf{Extract}(m,\mathsf{sk},r) = \mathsf{Resolve}(m, \mathsf{VESign}(m,\mathsf{sk},\mathsf{apk};r),\mathsf{ask},\mathsf{pk}).$$

By now numerous standard-model VES constructions have been proposed. However, not all of them are resolution-duplicate; especially not those having a randomized $\mathsf{Resolve}$ algorithm [CMSW14].

5.2 Revisiting Security: The Importance of Resolution Independence

In Boneh et al.'s original definition of a VES [BGLS03], the underlying signature scheme is required to be secure, in addition to the security properties of the encrypted signatures: completeness, unforgeability and opacity. Rückert and Schröder [RS09] added the properties of extractability and abuse freeness, and Calderon et al. [CMSW14] added the properties of resolution independence, but both omit (or are at least unclear about) the requirement that the underlying signature scheme be secure. Indeed, the latter paper says that they "additionally provide the adversary with access to the Sign oracle, as otherwise the underlying signature scheme could be completely broken and the VES would still be considered unforgeable." In fact, it can be completely broken anyway.

We will show that, with this omission, resolution independence is absolutely essential to not only the unforgeability, but even the correctness, of the underlying scheme. Resolution independence supplies the necessary glue to connect the security properties of the encrypted scheme to the underlying scheme. Contrapositively, we show that security including resolution independence is sufficient for the correctness and security of the underlying signature scheme, so that does not need to be proven separately. To be clear, we formally define what is meant by the underlying signature scheme.

Definition 5.9. *Let* $\mathsf{VES} = (\mathsf{AKeyGen}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{VESign}, \mathsf{VEVerify}, \mathsf{Resolve})$ *be a VES scheme. Then, we call* $\mathsf{Sig} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *the underlying signature scheme of* VES.

5.2.1 Counterexample

We now show that completeness, unforgeability, opacity, extractability and abuse freeness together do not imply the correctness or security of the underlying scheme.

Given a secure VES scheme $\mathsf{VES} = (\mathsf{AKeyGen}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{VESign}, \mathsf{VEVerify}, \mathsf{Resolve})$ for messages of length n, we now show how to derive a VES scheme $\mathsf{VES}' = (\mathsf{AKeyGen}, \mathsf{KeyGen}, \mathsf{Sign}', \mathsf{Verify}, \mathsf{VESign}, \mathsf{VEVerify}, \mathsf{Resolve})$ such that $\mathsf{Sign}'(m, \mathsf{sk})$ computes and outputs $\mathsf{Sign}(0^n, \mathsf{sk})$.

Theorem 5.10. *If* VES *is complete, unforgeable, opaque, extractable and abuse free, then so is* VES'.

Proof. The adversary in the unforgeability game must output a valid encrypted signature, but the set of valid encrypted signatures in VES and VES' are the same, and we have only weakened the oracles (by making Sign provide signatures only on 0^n), so unforgeability is preserved. The other properties do not mention the Sign algorithm at all, so they are unaffected. □

This scheme is intuitively both incorrect (as the signatures produced by Sign' cannot be verified) and insecure (as it gives away a forgery as soon as it is called

on a message other than 0^n). Nevertheless, VES' is secure as defined in [RS09], since their definition does not include the security of the underlying signature scheme. It is also much more catastrophically insecure than the separating example in [CMSW14, Section 3], which motivated the definition of resolution independence.

Theorem 5.11. *The underlying signature scheme* Sig *of* VES' *is neither correct nor secure.*

5.2.2 Filling the Gap

Lemma 5.12. *If* VES *is a complete and resolution-independent VES, then its underlying signature scheme* Sig *is correct.*

Proof. By completeness, for all $\kappa > 0$, all $(\mathsf{ask}, \mathsf{apk}) \xleftarrow{R} \mathsf{AKeyGen}(1^\kappa)$, all $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa)$ and all messages $m \in \mathcal{M}_{\mathsf{pk}}$, for $\omega \xleftarrow{R} \mathsf{VESign}(m, \mathsf{sk}, \mathsf{apk})$, with probability 1,

$$\mathsf{Verify}(m, \mathsf{Resolve}(m, \omega, \mathsf{ask}, \mathsf{pk}), \mathsf{pk}) = 1.$$

By resolution independence, $\mathsf{Resolve}(m, \omega, \mathsf{ask}, \mathsf{pk})$ is identically distributed to $\mathsf{Sign}(m, \mathsf{sk})$, so with probability 1,

$$\mathsf{Verify}(m, \mathsf{Sign}(m, \mathsf{sk}), \mathsf{pk}) = 1. \qquad \square$$

Lemma 5.13. *If* VES *is an opaque and resolution-independent VES, then its underlying signature scheme* Sig *is EUF-CMA secure.*

Proof. Let VES be a resolution-independent VES, and let Sig be the underlying signature scheme. We assume that there is an efficient adversary $\mathcal{A}$ breaking the EUF-CMA security of Sig with non-negligible probability, and construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to break the opacity of VES.

$\mathcal{B}$ takes as input an arbiter's public key apk and a signer's public key pk (with unknown corresponding private keys ask and sk), and passes pk as input to $\mathcal{A}$. Whenever $\mathcal{A}$ tries to query the Sign oracle on message m, $\mathcal{B}$ forwards m to its VESign oracle, obtaining $\omega = \mathsf{VESign}(m, \mathsf{sk}, \mathsf{apk})$; $\mathcal{B}$ then queries (m, ω) to its $\mathsf{Resolve}$ oracle, obtaining $\sigma = \mathsf{Resolve}(m, \mathsf{VESign}(m, \mathsf{sk}, \mathsf{apk}), \mathsf{ask}, \mathsf{pk})$, which it returns to $\mathcal{A}$. When $\mathcal{A}$ outputs (m^*, σ^*), $\mathcal{B}$ outputs the same.

By resolution independence, $\mathsf{Sign}(m, \mathsf{sk})$ and $\mathsf{Resolve}(m, \mathsf{VESign}(m, \mathsf{sk}, \mathsf{apk}), \mathsf{ask}, \mathsf{pk})$ are identically distributed, so we perfectly simulate $\mathcal{A}$'s Sign oracle.

If $\mathcal{A}$ never queried m^* to Sign, then $\mathcal{B}$ never queried m^* to $\mathsf{Resolve}$, and so $\mathcal{B}$ has the same non-negligible success probability as $\mathcal{A}$. $\square$

Theorem 5.14. *If a VES is complete, opaque and resolution independent, then its underlying signature scheme* Sig *is correct and secure.*

Proof. By Lemmas 5.12 and 5.13. $\square$

5.3 Verifiably Encrypted Signatures from SPS-EQ

Before, we give our VES construction from SPS-EQs, we introduce a new property characterizing SPS-EQs that come into question.

5.3.1 Perfectly Composing SPS-EQs

In the following, we require an SPS-EQ that satisfies the following property.

Definition 5.15 (Perfect composition)**.** *An SPS-EQ scheme* $\mathsf{SPS\text{-}EQ}$ *on* $(\mathbb{G}_i^*)^\ell$ allows perfect composition *if there exists an additional deterministic polynomial-time algorithm* $\mathsf{Switch}_\mathcal{R}$ *such that for all random tapes* $r \in \{0,1\}^*$ *and all tuples* $(\mathsf{sk}, \mathsf{pk}, M, \sigma, \mu)$ *with*

$$\mathsf{VKey}_\mathcal{R}(\mathsf{sk}, \mathsf{pk}) = 1 \qquad \sigma \leftarrow \mathsf{Sign}_\mathcal{R}(M, \mathsf{sk}; r) \qquad M \in (\mathbb{G}_i^*)^\ell \qquad \mu \in \mathbb{Z}_p^*$$

it holds that $(\mu M, \mathsf{Sign}_\mathcal{R}(\mu M, \mathsf{sk}; r)) = \mathsf{Switch}_\mathcal{R}(M, \sigma, \mu, \mathsf{pk})$.

Intuitively, algorithm $\mathsf{Switch}_\mathcal{R}$ updates only those parts of σ that are affected by updating the representative from M to μM, where the randomness inside σ remains to be that of the initial signing process.

We are now going to see that Scheme 2 fulfills Definition 5.15.

Lemma 5.16. *Scheme 2 allows perfect composition.*

Proof (Sketch). Let $\mathsf{Switch}_\mathcal{R}$ be the algorithm that arises from the $\mathsf{ChgRep}_\mathcal{R}$ algorithm of Scheme 2 when fixing its internally drawn randomizer ψ to 1, i.e.,

$$\mathsf{Switch}_\mathcal{R}(M, \sigma, \mu, \mathsf{pk}) := \mathsf{ChgRep}_\mathcal{R}(M, \sigma, \mu, \mathsf{pk}; (\psi = 1)).$$

Let $\kappa > 0$, $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P}) \xleftarrow{R} \mathsf{BGGen}_\mathcal{R}(1^\kappa)$, $\ell > 1$. Further, let $r \in \{0,1\}^*$, $(\mathsf{sk}, \mathsf{pk})$, $M \in (\mathbb{G}_1^*)^\ell$, and σ be such that $\mathsf{VKey}_\mathcal{R}(\mathsf{sk}, \mathsf{pk}) = 1$ and $\sigma \leftarrow \mathsf{Sign}_\mathcal{R}(M, \mathsf{sk}; r)$ (internally drawing $y \in \mathbb{Z}_p^*$ using r).

Then, $\sigma = (Z, Y, \hat{Y}) \in \mathbb{G}_1 \times \mathbb{G}_1^* \times \mathbb{G}_2^*$ is of the form $(Z = y \sum x_i M_i, Y = \frac{1}{y} P$, $\hat{Y} = \frac{1}{y}\hat{P})$. Executing $\mathsf{ChgRep}_\mathcal{R}(M, \sigma, \mu, \mathsf{pk}; (\psi = 1))$ for $\mu \in \mathbb{Z}_p^*$ gives $(\mu M, \sigma')$ such that $\sigma' = (Z', Y', \hat{Y}') = (\mu Z, Y, \hat{Y}) = (Z = \psi y \sum x_i \mu M_i, Y = \frac{1}{\psi y} P, \hat{Y} = \frac{1}{\psi y}\hat{P}) = (Z = y \sum x_i \mu M_i, Y = \frac{1}{y} P, \hat{Y} = \frac{1}{y}\hat{P})$.

Next, observe that $\mathsf{Sign}_\mathcal{R}(\mu M, \mathsf{sk}; r)$ internally draws $y \xleftarrow{R} \mathbb{Z}_p^*$ using r (i.e., the same y as above), returns $\sigma'' = (Z'', Y'', \hat{Y}'')$ with $Z'' = y \sum x_i \mu M_i$, $Y = \frac{1}{y} P$ and $\hat{Y} = \frac{1}{y}\hat{P}$ giving the same signature on μM as $\mathsf{Switch}_\mathcal{R}(M, \sigma, \mu, \mathsf{pk})$[2]. □

Remark 5.17. Observe that Scheme 3 does not support perfect composition: Let $\sigma = (Z, Y, \hat{Y}, R_1, R_2)$ be a signature on M produced by Scheme 3. Then, the reason for this are the *random* signature values R_1, R_2, which do not stay the same but, instead, are being multiplied by μ when performing a change of representative from M to μM for some $\mu \in \mathbb{Z}_p^*$.

5.3.2 The Construction

In Scheme 6, we show how a VES can be built black-box from any SPS-EQ construction SPS-EQ that allows perfect composition and has a deterministic bilinear-group generation algorithm $\mathsf{BGGen}_{\mathcal{R}}$.[1] The latter is necessary as both key generation algorithms AKeyGen and KeyGen on input 1^κ have to output key pairs with respect to the same bilinear group BG.

Scheme 6 A VES construction from SPS-EQ.

$\mathsf{AKeyGen}(1^\kappa)$: Given a security parameter 1^κ, compute $\mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$, pick $a \xleftarrow{R} \mathbb{Z}_p^*$, compute $A \leftarrow aP$ and output $(\mathsf{ask}, \mathsf{apk}) \leftarrow (a, (\mathsf{BG}, A))$.

$\mathsf{KeyGen}(1^\kappa)$: Given a security parameter 1^κ, compute $\mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$ and output $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^\ell)$ for $\ell = 3$.

$\mathsf{Sign}(m, \mathsf{sk}; (r_1, r_2))$: Given a message $m \in \mathbb{Z}_p^*$, secret key sk and two random tapes $r_1, r_2 \in \{0,1\}^*$, pick $s \xleftarrow{R} \mathbb{Z}_p^*$ using r_1 and compute $\sigma' \leftarrow \mathsf{Sign}_{\mathcal{R}}((msP, sP, P), \mathsf{sk}; r_2)$ using randomness r_2. Finally, output $\sigma \leftarrow (\sigma', sP)$.

$\mathsf{Verify}(m, \sigma, \mathsf{pk})$: Given a message $m \in \mathbb{Z}_p^*$, a signature $\sigma = (\sigma', S)$ and a public key pk, output whatever $\mathsf{Verify}_{\mathcal{R}}((mS, S, P), \sigma', \mathsf{pk})$ outputs.

$\mathsf{VESign}(m, \mathsf{sk}, \mathsf{apk}; (r_1, r_2))$: Given a message $m \in \mathbb{Z}_p^*$, secret key sk, the arbiter public key $\mathsf{apk} = A$ and two random tapes $r_1, r_2 \in \{0,1\}^*$, pick $s \xleftarrow{R} \mathbb{Z}_p^*$ using r_1 and compute $\omega' \leftarrow \mathsf{Sign}_{\mathcal{R}}((msA, sA, A), \mathsf{sk}; r_2)$ using randomness r_2. Finally, output $\omega \leftarrow (\omega', sA)$.

$\mathsf{VEVerify}(m, \omega, \mathsf{pk}, \mathsf{apk})$: Given a message $m \in \mathbb{Z}_p^*$, an encrypted signature $\omega = (\omega', W)$, a public key pk and an arbiter public key $\mathsf{apk} = A$, output whatever $\mathsf{Verify}_{\mathcal{R}}((mW, W, A), \omega', \mathsf{pk})$ outputs.

$\mathsf{Resolve}(m, \omega, \mathsf{ask}, \mathsf{pk})$: Given a message $m \in \mathbb{Z}_p^*$, an encrypted signature $\omega = (\omega', sA)$, a public key pk and an arbiter secret key $\mathsf{ask} \leftarrow a$, check whether $\mathsf{VEVerify}(m, \omega, \mathsf{pk}, \mathsf{apk}) = 0$ and return $\perp$ if so. Otherwise, compute $((msP, sP, P), \sigma') \leftarrow \mathsf{Switch}_{\mathcal{R}}((msA, sA, A), \omega, \frac{1}{a}, \mathsf{pk})$ and output $\sigma \leftarrow (\sigma', sP)$.

Remark 5.18. Observe that, independently of the instantiation of Scheme 6 with a concrete SPS-EQ, the efficiency of the Verify respectively VEVerify can be improved by precomputing parts of the PPEs that solely depend on P and

[1] As already pointed out in Chapter 4, this is, e.g., the case for BN curves [BN06]; the most common choice for Type-3 pairings.

pk respectively A and pk, and including the resulting $\mathbb{G}_T$ elements into (the updated) user public key pk.

In the following, we are going to analyze the security of Scheme 6 and prove unforgeability, opacity and abuse freeness as well as resolution duplication. Completeness follows straightforwardly from inspection.

Theorem 5.19. *The VES in Scheme 6 is unforgeable given that* SPS-EQ *is unforgeable.*

Proof. We assume that there is an efficient adversary $\mathcal{A}$ winning the unforgeability game with non-negligible probability; then we are able to construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to break the EUF-CMA security of the underlying SPS-EQ scheme with non-negligible probability.

$\mathcal{B}$ obtains $\mathsf{pk}_{\mathcal{R}}$ of the SPS-EQ scheme SPS-EQ with $\ell = 3$ (and thereby implicitly the bilinear group BG) from the challenger $\mathcal{C}$ of the EUF-CMA security game, and sets $(\mathsf{sk}, \mathsf{pk}) \leftarrow (\perp, \mathsf{pk}_{\mathcal{R}})$. Then $\mathcal{B}$ picks $a \xleftarrow{R} \mathbb{Z}_p^*$, computes $A \leftarrow aP$ and sets $(\mathsf{ask}, \mathsf{apk}) \leftarrow (a, (\mathsf{BG}, A))$. Next, $\mathcal{B}$ sets up a list $L \leftarrow \emptyset$ to keep track of representatives queried to $\mathcal{C}$, runs $\mathcal{A}$ on $(\mathsf{pk}, \mathsf{apk})$ and answers $\mathcal{A}$'s oracle queries to the Resolve oracle as in the real game and simulates queries to all other oracles as follows:

$\mathsf{Sign}(\cdot, \mathsf{sk})$: If $\mathcal{A}$ submits a query for $m \in \mathbb{Z}_p^*$, $\mathcal{B}$ queries $\mathcal{C}$'s signing oracle for the message (msP, sP, P) for $s \xleftarrow{R} \mathbb{Z}_p^*$, gets in return a corresponding signature σ', appends $\{(msA, sA, A)\}$ to $L[m]$ and outputs the plain signature $\sigma \leftarrow (\sigma', sP)$.

$\mathsf{VESign}(\cdot, \mathsf{sk}, \mathsf{apk})$: If $\mathcal{A}$ submits a query for $m \in \mathbb{Z}_p^*$, $\mathcal{B}$ queries $\mathcal{C}$'s signing oracle for the message (msA, sA, A) for $s \xleftarrow{R} \mathbb{Z}_p^*$, gets in return a corresponding signature ω', appends $\{(msA, sA, A)\}$ to $L[m]$ and outputs the encrypted signature $\omega \leftarrow (\omega', sA)$.

If at some point $\mathcal{A}$ outputs a valid encrypted message-signature pair $(m^*, \omega^* = (\omega'^*, W^*))$, such that it has not previously queried m^* to any of the oracles, then $\mathcal{B}$ will output $((m^*W^*, W^*, A), \omega'^*)$ to $\mathcal{C}$.

Note that the distribution of all values returned to $\mathcal{A}$ during the simulation is identical to the distribution of these values during a real game.

By construction, $((m^*W^*, W^*, A), \omega'^*)$ constitutes a valid message-signature pair. It remains to be shown that for $M^* = (m^*W^*, W^*, A)$, the class $[M^*]_{\mathcal{R}}$ is different from all classes represented by elements in L, if m^* is different from all messages queried to the oracles. VEVerify demands that the third vector component of M^* be A, which uniquely determines the representative for each class and allows for comparison. Now, if there is some $M_i = (m_iW_i, W_i, A) \in L$ queried to the VESign or the Sign oracle coinciding with M^* in the second component, then both vectors still differ in the first component for $m^* \neq m_i$. Likewise, if they coincide in the first component for $m^* \neq m_i$, then they cannot have equal second components. Hence, $M^* \neq M_i$ for any M_i in L. □

Theorem 5.20. *The VES in Scheme 6 is opaque given that the DHI assumption holds in* $\mathbb{G}_1$ *and that* SPS-EQ *is unforgeable and allows perfect composition.*

Proof. We assume that there is an efficient adversary $\mathcal{A}$ winning the opacity game with non-negligible probability. Then we are able to construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ either to break with non-negligible probability the EUF-CMA security of the underlying SPS-EQ scheme SPS-EQ (*Type-1* adversary) if $\mathcal{A}$ has neither queried to the VESign nor to the Resolve oracle for m^*; or to break the DHI assumption (*Type-2* adversary) if $\mathcal{A}$ has only queried to the VESign but not to the Resolve oracle for m^*.

In the following, $\mathcal{B}$ guesses $\mathcal{A}$'s strategy, i.e., the type of forgery $\mathcal{A}$ will conduct. We are now going to describe the setup, the initialization of the environment, the reduction and the abort conditions for each type.

Type 1: $\mathcal{B}$ obtains $\mathsf{pk}_{\mathcal{R}}$ of the SPS-EQ scheme SPS-EQ with $\ell = 3$ (and thereby implicitly the bilinear group BG) from the challenger $\mathcal{C}$ of the EUF-CMA security game and sets $(\mathsf{sk}, \mathsf{pk}) \leftarrow (\perp, \mathsf{pk}_{\mathcal{R}})$. Furthermore, $\mathcal{B}$ picks $a \xleftarrow{R} \mathbb{Z}_p^*$, computes $A \leftarrow aP$ and sets $(\mathsf{ask}, \mathsf{apk}) \leftarrow (a, (\mathsf{BG}, A))$. Next, $\mathcal{B}$ runs $\mathcal{A}$ on $(\mathsf{pk}, \mathsf{apk})$ and answers $\mathcal{A}$'s oracle queries to the Resolve oracle as in a real game and simulates queries to the VESign oracle as follows:

VESign$(\cdot, \mathsf{sk}, \mathsf{apk})$: If $\mathcal{A}$ submits a query for $m \in \mathbb{Z}_p^*$, $\mathcal{B}$ queries $\mathcal{C}$'s signing oracle for the message (msA, sA, A) for $s \xleftarrow{R} \mathbb{Z}_p^*$, then $\mathcal{B}$ gets in return a signature ω' and outputs (ω', sA).

If at some point $\mathcal{A}$ outputs a valid message-signature pair (m^*, σ^*) with $\sigma^* = (\sigma'^*, S^*)$ and has queried to the VESign oracle for m^*, $\mathcal{B}$ will abort (Type-2 forgery). Else, $\mathcal{B}$ will output $((m^*S^*, S^*, P), \sigma'^*)$ to $\mathcal{C}$.

Note that the distribution of all values returned to $\mathcal{A}$ during the simulation is identical to the distribution of these values during a real game, which makes the simulation perfect.

By construction, $((m^*S^*, S^*, P), \sigma'^*)$ constitutes a valid SPS-EQ message-signature pair. It remains to be shown that for $M^* = (m^*S^*, S^*, P)$, the class $[M^*]_{\mathcal{R}}$ is different from all classes queried to $\mathcal{C}$, if m^* is different from all messages queried to the VESign oracle. Verify demands that the third vector component of M^* be P, which uniquely determines the representative for each class and allows for comparison. Now, if there is some $M_i = (m_iS_i, S_i, P)$ coinciding with M^* in the second component, then both vectors still differ in the first component for $m^* \neq m_i$. Likewise, if they coincide in the first component for $m^* \neq m_i$, then they cannot have equal second components. Hence, $M^* \neq M_i$ for any M_i queried to $\mathcal{C}$.

Type 2: We assume q_r to be the number of queries to the Resolve oracle and, w.l.o.g., we assume that $\mathcal{A}$ does not query to the Resolve oracle for the same message twice.

$\mathcal{B}$ obtains an instance $(\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P}), aP)$ of the DHI problem in $\mathbb{G}_1$ from the challenger $\mathcal{C}$ and fixes an index $j \xleftarrow{R} [q_r + 1]$. $\mathcal{B}$ executes

$(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^{\ell})$ for $\ell = 3$, sets $\mathsf{ask} \leftarrow \perp$, runs $\mathcal{A}$ on $(\mathsf{pk}, \mathsf{apk} \leftarrow (\mathsf{BG}, A = aP))$, sets up a list $L \leftarrow \emptyset$ and simulates queries to the oracles as follows:

$\mathsf{VESign}(\cdot, \mathsf{sk}, \mathsf{apk})$: On the j'th query, $\mathcal{B}$ sets a bit $b \leftarrow [j = j']$, picks $s \xleftarrow{R} \mathbb{Z}_p^*$, sets $W \leftarrow s(A + bP)$, runs $\omega' \xleftarrow{R} \mathsf{Sign}_{\mathcal{R}}((msW, sW, A), \mathsf{sk}; r_2)$, appends (m, s, r_2) to L and returns the encrypted signature $\omega \leftarrow (\omega', W)$.

$\mathsf{Resolve}(\cdot, \cdot, \mathsf{ask}, \mathsf{pk})$: If $\mathcal{A}$ submits a query for $m \in \mathbb{Z}_p^*$ and ω, then $\mathcal{B}$ checks whether $\mathsf{VEVerify}(m, \omega, \mathsf{pk}, \mathsf{apk}) = 1$ and returns $\perp$ if this is not the case.

Otherwise, it determines the index j' such that $L[j'][1] = m$. If $j = j'$, then $\mathcal{B}$ aborts. Let $L[j'] = (m, s, r_2)$. $\mathcal{B}$ computes $\sigma' \leftarrow \mathsf{Sign}_{\mathcal{R}}((msP, sP, P), \mathsf{sk}; r_2)$ and returns the plain signature $\sigma \leftarrow (\sigma', sP)$.

If at some point $\mathcal{A}$ outputs a valid message-signature pair $(m^*, \sigma^* = (\sigma'^*, S^*))$, then $\mathcal{B}$ determines the index j' such that $L[j'][1] = m^*$. If there is no such index j', then $\mathcal{B}$ aborts (Type-1 forgery). Otherwise, if $j' \neq j$, then $\mathcal{B}$ aborts as well. Else, $\mathcal{B}$ retrieves $s \leftarrow L[j][2]$ and outputs $\frac{1}{s}S^* - P = \frac{1}{a}P$ as a solution to the DHI problem.

If SPS-EQ allows perfect composition, the distribution of all values returned to $\mathcal{A}$ during the simulation is identical to the distribution of these values during a real game, which makes the simulation perfect (as it guarantees that $\mathsf{Sign}_{\mathcal{R}}((msW, sW, A), \mathsf{sk}; r_2) = \omega' = \mathsf{Switch}_{\mathcal{R}}((msP, sP, P), \sigma', a, \mathsf{pk})$ for $\sigma' \leftarrow \mathsf{Sign}_{\mathcal{R}}((msP, sP, P), \mathsf{sk}; r_2)$ during the simulation of both oracles).

Finally, we consider $\mathcal{B}$'s success probability. We assume $\mathcal{A}$ to be a Type-2 adversary that is able to break the opacity of the scheme with probability $\epsilon(\kappa)$. Then, $\mathcal{B}$ does not abort the simulation with probability at least $(1 - \frac{1}{q_r+1})^{q_r} \geq \frac{1}{\exp(1)}$ and $\mathcal{A}$ uses the jth query for the forgery with probability at least $\frac{1}{q_r+1}$. Hence, it holds that $\epsilon(\kappa) \leq \exp(1) \cdot (q_r + 1) \cdot \epsilon_{DHI}(\kappa)$, where $\epsilon_{DHI}(\kappa)$ is the advantage of solving the DHI assumption. □

Note that for any SPS-EQ over $\mathbb{G}_i$ we assume the DDH assumption to hold in $\mathbb{G}_i$, which implies the CDH assumption. Moreover, the CDH assumption is equivalent to the DHI assumption [BDZ03] (as already pointed out in Chapter 2). Therefore, opacity is black-box from any perfectly composing SPS-EQ over $\mathbb{G}_i$.

Theorem 5.21. *The VES in Scheme 6 is unconditionally extractable.*

Proof (Sketch). This immediately follows from the SPS-EQ correctness property: Observe that for an unresolved tuple (m, ω) with $\omega = (\omega', sA)$ it holds that $\mathsf{Verify}_{\mathcal{R}}((msA, sA, A), \omega', \mathsf{pk}) = 1$ if and only if $\mathsf{Verify}_{\mathcal{R}}((msP, sP, P), \sigma', \mathsf{pk}) = 1$, where $((msP, sP, P), \sigma') \leftarrow \mathsf{Switch}_{\mathcal{R}}((msA, sA, A), \omega', \frac{1}{a}, \mathsf{pk})$, since

$$[(msA, sA, A)]_{\mathcal{R}} = [(msP, sP, P)]_{\mathcal{R}}.$$

□

Theorem 5.22. *The VES in Scheme 6 is abuse free given that* SPS-EQ *is unforgeable.*

Proof. We assume that there is an efficient adversary $\mathcal{A}$ winning the abuse freeness game with non-negligible probability; then we are able to construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to break the EUF-CMA security of SPS-EQ with non-negligible probability.

$\mathcal{B}$ obtains $\mathsf{pk}_{\mathcal{R}}$ of the SPS-EQ scheme SPS-EQ with $\ell = 3$ (and thereby implicitly the bilinear group BG) from the challenger $\mathcal{C}$ of the EUF-CMA security game, sets $(\mathsf{sk}, \mathsf{pk}) \leftarrow (\perp, \mathsf{pk}_{\mathcal{R}})$. Furthermore, $\mathcal{B}$ picks $a \xleftarrow{R} \mathbb{Z}_p^*$, computes $A \leftarrow aP$ and sets $(\mathsf{ask}, \mathsf{apk}) = (a, (\mathsf{BG}, A))$. Next, $\mathcal{B}$ runs $\mathcal{A}$ on $(\mathsf{pk}, \mathsf{ask}, \mathsf{apk})$ and answers $\mathcal{A}$'s oracle queries as follows:

$\mathsf{VESign}(\cdot, \mathsf{sk}, \mathsf{apk})$: If $\mathcal{A}$ submits a query for $m \in \mathbb{Z}_p^*$, $\mathcal{B}$ queries to $\mathcal{C}$'s signing oracle for the message $(m \cdot sA, sA, A)$ for $s \xleftarrow{R} \mathbb{Z}_p^*$, gets in return a corresponding signature ω' and outputs the encrypted signature $\omega \leftarrow (\omega', sA)$.

If at some point $\mathcal{A}$ outputs a valid encrypted message-signature pair $(m^*, \omega^* = (\omega'^*, W^*))$, such that it has not previously queried m^* to the VESign oracle, then $\mathcal{B}$ will output $((m^*W^*, W^*, A), \omega'^*)$ to $\mathcal{C}$; otherwise, $\mathcal{B}$ will abort.

Note that the distribution of all values returned to $\mathcal{A}$ during the simulation is identical to the distribution of these values during a real game.

By construction, $((m^*W^*, W^*, A), \omega'^*)$ constitutes a valid message-signature pair. It remains to be shown that for $M^* = (m^*W^*, W^*, A)$, the class $[M^*]_{\mathcal{R}}$ is different from all classes queried to $\mathcal{C}$, if m^* is different from all messages queried to the VESign oracle. VEVerify demands that the third vector component of M^* be A, which uniquely determines the representative for each class and allows for comparison. Now, if there is some $M_i = (m_i \cdot W_i, W_i, A)$ coinciding with M^* in the second component, then both vectors still differ in the first component for $m^* \neq m_i$. Likewise, if they coincide in the first component for $m^* \neq m_i$, then they cannot have equal second components. Hence, assuming that $m^* \neq m_i$ and $M^* = M_i$ for some M_i queried to $\mathcal{C}$, immediately gives a contradiction. □

The following theorem states that Scheme 6 is resolution duplicate given that the underlying SPS-EQ allows perfect composition (i.e., fulfills Definition 5.15). In particular, it is resolution independent, the importance of which was established in Section 5.2. It will allow also us to derive a PKE scheme (cf. Section 5.4). Note that Scheme 2 fulfills Definition 5.15.

Theorem 5.23. *The VES in Scheme 6 is resolution duplicate given that* SPS-EQ *allows perfect composition.*

Proof. Here, we have to show (1) that the outputs of $\mathsf{Resolve}(m, \mathsf{VESign}(m, \mathsf{sk}, \mathsf{apk}), \mathsf{ask}, \mathsf{pk})$ and $\mathsf{Sign}(m, \mathsf{sk})$ are distributed identically, (2) that Resolve is deterministic and (3) that there exists a PPT algorithm $\mathsf{Extract}(\cdot, \cdot, \cdot)$, such that for all $(\mathsf{ask}, \mathsf{apk}) \xleftarrow{R} \mathsf{AKeyGen}(1^\kappa)$, $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa), m \in \mathcal{M}_{\mathsf{pk}}$, and random tapes $r \in \{0,1\}^*$, it is the case that

$$\mathsf{Extract}(m, \mathsf{sk}, r) = \mathsf{Resolve}(m, \mathsf{VESign}(m, \mathsf{sk}, \mathsf{apk}; r), \mathsf{ask}, \mathsf{pk}).$$

Property (2) is easy to see, since Resolve internally runs algorithm $\mathsf{Switch}_{\mathcal{R}}$, which is deterministic. All other parts of Resolve are deterministic as well.

The extract algorithm for Property (3) can be specified as $\mathsf{Extract}(m, \mathsf{sk}, r) := \mathsf{Sign}(m, \mathsf{sk}; r) = \mathsf{Sign}(m, \mathsf{sk}; (r_1, r_2)) = (\mathsf{Sign}_{\mathcal{R}}((msP, sP, P), \mathsf{sk}; r_2), sP)$ with s drawn uniformly from $\mathbb{Z}_p^*$ using random coins r_1. For the RHS, we have

$$\mathsf{Resolve}(m, \mathsf{VESign}(m, \mathsf{sk}, \mathsf{apk}; r_2), \mathsf{ask}, \mathsf{pk}) =$$
$$\mathsf{Resolve}(m, (\mathsf{Sign}_{\mathcal{R}}((msA, sA, A), \mathsf{sk}; r_2), sA), \mathsf{ask}, \mathsf{pk}) =$$
$$(\mathsf{Switch}_{\mathcal{R}}((msA, sA, A), \mathsf{Sign}_{\mathcal{R}}((msA, sA, A), \mathsf{sk}; r_2), a^{-1}, \mathsf{pk})[2], sP),$$

where s and t are as above. If SPS-EQ allows perfect composition, this gives the same output as the specified Extract algorithm.

With regard to (1) observe that Property (3) and the fact that the Extract algorithm can be expressed by Sign implies that the output distributions of $\mathsf{Sign}(m, \mathsf{sk})$ and $\mathsf{Resolve}(m, \mathsf{VESign}(m, \mathsf{sk}, \mathsf{apk}), \mathsf{ask}, \mathsf{pk})$ are identical. □

Discussion

In sum, Scheme 6 is a resolution-duplicate VES constructed black-box from any perfectly composing SPS-EQ. Observe that for all our proofs, we only made use of the DHI assumption and the SPS-EQ properties correctness, unforgeability and perfect composition. Hence, we could in fact use a weaker notion of SPS-EQs, which does not require the indistinguishability of classes (and, thus, the DDH assumption to hold in $\mathbb{G}_1$). In particular, we could as well employ an SPS-EQ scheme (e.g., a Type-1 SPS-EQ scheme) that provides the same kind of malleablity, while only requiring the CDH assumption to hold in $\mathbb{G}_1$ (which we always assume to hold in cryptographically strong groups). This could widen the range of potential standard-model instantiations of Scheme 6.

Another measure to this end is the following relaxation. To just build a resolution-independent VES, we can simply replace the quite strong perfect composition property by perfect adaptation (Definition 3.7). In doing so, we need to replace the execution of $\mathsf{Switch}_{\mathcal{R}}((msA, sA, A), \omega, \frac{1}{a}, \mathsf{pk})$ inside Resolve by $\mathsf{ChgRep}_{\mathcal{R}}((msA, sA, A), \omega, \frac{1}{a}, \mathsf{pk})$. Recall that perfect adaptation ensures that signatures output by $\mathsf{ChgRep}_{\mathcal{R}}$ are distributed like fresh signatures. Therefore, this immediately implies a resolution-independent VES construction: Resolution independence demands that resolved encrypted signatures are distributed like plain signatures and resolution essentially relies on changing the representative using $\mathsf{ChgRep}_{\mathcal{R}}$ (as outlined above). Even so, our focus here was to build a resolution-duplicate VES from SPS-EQ, since it shows the following non-trivial relation between SPS-EQ and PKE.

5.4 Public-Key Encryption from SPS-EQ

In this section, we show how to convert any SPS-EQ satisfying perfect composition (Definition 5.15) into a PKE scheme. This connection is somewhat surprising, as it is well known that regular signature schemes do not imply PKE (in a black-box way). However, there is no contradiction as SPS-EQs have more structure than regular signature schemes.

The basic idea is to instantiate the transformation given by Calderon et al. [CMSW14]. This transformation turns any secure, resolution-duplicate VES scheme into a PKE scheme, in a black-box way. We have already shown how to construct a secure VES scheme and that it is resolution duplicate, in Section 5.3. The basic idea of the transformation is an application of the Goldreich-Levin trick [GL89] to the setting of VES. That is, we view $\langle \sigma, r \rangle$ as the hard-core predicate for VESign, i.e., given ω and r it should be hard to predict the value of $\langle \sigma, r \rangle$ (cf. Section 2.1.2). This intuition is formally shown in the following lemma.

Lemma 5.24. *Let* VES *be a VES and* $b(x, r) := \langle x, r \rangle$ *for any* $x \in \{0,1\}^*$ *and* $r \in \{0,1\}^*$ *such that* $|x| = |r|$*. Then, if* VES *is opaque for all choices of* $(\mathsf{ask}, \mathsf{apk}) \xleftarrow{R} \mathsf{AKeyGen}(1^\kappa)$*, all choices of* $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa)$ *and all messages* $m \in \mathcal{M}_{\mathsf{pk}}$*, it is hard to compute the value* $b(\sigma, r)$ *with probability significantly greater than* $1/2$ *when given* $m, \mathsf{apk}, \mathsf{pk}, \omega \xleftarrow{R} \mathsf{VESign}(m, \mathsf{sk}, \mathsf{apk})$ *and* $r \xleftarrow{R} \{0,1\}^{|\sigma|}$*, where* $\sigma \leftarrow \mathsf{Resolve}(\omega, \mathsf{ask}, \mathsf{pk})$*.*

The proof is given in [CMSW14] and closely follows that of Goldreich [Gol01]. It leads to the construction of a CPA-secure PKE scheme in Scheme 7.

Scheme 7 A PKE scheme from resolution-duplicate VESs.

$\mathsf{KeyGen}(1^\kappa)$: Given a security parameter 1^κ, output $(\mathsf{ask}, \mathsf{apk}) \xleftarrow{R} \mathsf{AKeyGen}(1^\kappa)$.

$\mathsf{Enc}(m, \mathsf{apk})$: Given plaintext $m \in \{0,1\}$ and public key apk, generate a key pair $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa)$, pick a random tape r, compute a VES $\omega \leftarrow \mathsf{VESign}(0, \mathsf{sk}, \mathsf{apk}; r)$, $\sigma \leftarrow \mathsf{Extract}(m, \mathsf{sk}, r)$, pick $r_\sigma \xleftarrow{R} \{0,1\}^{|\sigma|}$ and set $c_0 \leftarrow m \oplus \langle \sigma, r_\sigma \rangle$. Finally, output ciphertext $c \leftarrow (\mathsf{pk}, \omega, r_\sigma, c_0)$.

$\mathsf{Dec}(c, \mathsf{ask})$: Given ciphertext c and secret key ask, parse c as $(\mathsf{pk}, \omega, r_\sigma, c_0)$ and return $\perp$ if $\mathsf{VEVerify}(0, \omega, \mathsf{pk}, \mathsf{apk}) = 0$. Otherwise, output $m \leftarrow c_0 \oplus \langle \sigma, r_\sigma \rangle$ with $\sigma \leftarrow \mathsf{Resolve}(0, \mathsf{pk}, \omega, \mathsf{ask}, \mathsf{pk})$.

Regarding security, it was shown in [CMSW14] that the above construction is correct and, most of all, CPA-secure:

Theorem 5.25. *If* VES *is a resolution-duplicate and opaque VES, then Scheme 7 is IND-CPA secure.*

The following corollary points out the relation of perfectly composing SPS-EQs to PKE.

Corollary 5.26. *Let* $\mathsf{SPS\text{-}EQ}$ *be an EUF-CMA secure and perfectly composing SPS-EQ and* VES *be the VES in Scheme 6 instantiated with* $\mathsf{SPS\text{-}EQ}$*. Then, Scheme 7 instantiated with* VES *is IND-CPA secure.*

6
Set Commitments

The user's going to pick dancing pigs over security every time.
— Bruce Schneier

In this chapter, we introduce a new commitment type which allows for committing to sets and, besides ordinary openings, also supports openings of subsets. After formalizing the primitive, we give an efficient construction with succinct commitments and openings.

In [KZG10], Kate et al. introduced the notion of constant-size polynomial commitments. They present two schemes where one is computationally hiding and the other one is perfectly hiding.

Following a similar approach, we construct perfectly hiding set commitments, which allow us to commit to a set $S \subset \mathbb{Z}_p$, by committing to a monic polynomial whose roots are the elements of S. A feature we are aiming at is the possibility to open arbitrary subsets of the committed set, which is implicitly done by opening non-trivial factors of the committed polynomial.

The results in this paper are related to the polynomial-commitment scheme given in [HS14] but further try to abstract them by only considering sets. These results are part of [FHS16] which is currently in review.

6.1 Definitions

We start with discussing the abstract model and the security properties of our set-commitment scheme.

Definition 6.1 (Set commitment (SC) scheme)**.** *An* SC scheme SC *consists of the following PPT algorithms.*

$\mathsf{Setup}(1^\kappa, 1^t)$: *This probabilistic algorithm takes input a security parameter* κ *and an upper bound for the set cardinality* $t \in \mathbb{N}$*, both in unary form. It outputs public parameters* pp *(including a description of an efficiently samplable message space* $\mathcal{S}_{\mathsf{pp}}$ *containing sets of maximum cardinality* t*).*

$\mathsf{Commit}(\mathsf{pp}, S)$: *This probabilistic algorithm takes input the public parameters* pp *defining message space* $\mathcal{S}_{\mathsf{pp}}$ *and a non-empty set* $S \in \mathcal{S}_{\mathsf{pp}}$*. It outputs a commitment* C *to set* S *and opening* O*.*

$\mathsf{Open}(\mathsf{pp}, C, O)$: *This deterministic algorithm takes input the public parameters* pp*, a commitment* C *and opening* O*. It outputs* S *if* O *is a valid opening of* C *for* $S \in \mathcal{S}_{\mathsf{pp}}$ *and* $\perp$ *otherwise.*

$\mathsf{OpenSubset}(\mathsf{pp}, C, O, T)$: *This (deterministic) algorithm takes input the public parameters* pp*, a commitment* C*, opening* O *for set* $S \in \mathcal{S}_{\mathsf{pp}}$ *and a non-empty set* T*. It returns* $\perp$ *if* $T \not\subseteq S$*; else it returns a witness* W *for* T *being a subset of* S*.*

$\mathsf{VerifySubset}(\mathsf{pp}, C, T, W)$: *This deterministic algorithm takes input the public parameters* pp*, a commitment* C*, a non-empty set* T *and a witness* W*. It verifies whether* W *is a witness for* T *being a subset of the set committed to in* C*, in which case it outputs* 1*, and* 0 *otherwise.*

We call a set-commitment scheme *secure*, if it is *correct*, *binding*, *subset-sound* and *hiding*. The properties are as follows, where the definitions of correctness, binding and hiding are mostly straightforward.

Definition 6.2 (Correctness)**.** *An SC scheme* SC *is* correct *if for all* $t > 0$*, all* $\kappa > 0$*, all choices of* $\mathsf{pp} \xleftarrow{R} \mathsf{Setup}(1^\kappa, 1^t)$*, all* $S \in \mathcal{S}_{\mathsf{pp}}$ *and all non-empty* $T \subseteq S$ *the following holds:*

1. $\Pr\left[\, \mathsf{Open}(\mathsf{pp}, \mathsf{Commit}(\mathsf{pp}, S)) = S \,\right] = 1.$
2. $$\Pr\left[\begin{array}{l} (C, O) \xleftarrow{R} \mathsf{Commit}(\mathsf{pp}, S), \\ W \xleftarrow{R} \mathsf{OpenSubset}(\mathsf{pp}, C, O, T) \end{array} : \mathsf{VerifySubset}(\mathsf{pp}, C, T, W) = 1 \right] = 1.$$

Definition 6.3 (Binding)**.** *An SC scheme* SC *is* binding *if for all* $t > 0$ *and all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l} \mathsf{pp} \xleftarrow{R} \mathsf{Setup}(1^\kappa, 1^t), \\ (C, O, O') \xleftarrow{R} \mathcal{A}(\mathsf{pp}), \\ S \leftarrow \mathsf{Open}(\mathsf{pp}, C, O), \\ S' \leftarrow \mathsf{Open}(\mathsf{pp}, C, O') \end{array} : \quad S \neq S' \quad \wedge \quad S, S' \neq \perp \right] \leq \epsilon(\kappa).$$

Subset soundness requires it to be infeasible to perform subset openings using non-subsets.

Definition 6.4 (Subset soundness)**.** *An SC scheme* SC *is* subset-sound *if for all* $t > 0$ *and all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l} \mathsf{pp} \xleftarrow{R} \mathsf{Setup}(1^\kappa, 1^t), \\ (C, O, T, W) \xleftarrow{R} \mathcal{A}(\mathsf{pp}), \\ S \leftarrow \mathsf{Open}(\mathsf{pp}, C, O) \end{array} : \begin{array}{r} S \neq \perp \quad \wedge \quad T \not\subseteq S \\ \wedge \quad \mathsf{VerifySubset}(\mathsf{pp}, C, T, W) = 1 \end{array} \right] \leq \epsilon(\kappa).$$

Hiding resembles the conventional hiding definition for commitments (cf. Definition 2.23), but, in addition, the adversary is given access to an $\mathsf{OpenSubset}$ oracle for subsets of the intersection of the two challenge sets.

Definition 6.5 (Hiding)**.** *An SC scheme* SC *is* hiding *if for all* $t > 0$ *and all PPT adversaries* $\mathcal{A}$ *with oracle access to* $\mathsf{OpenSubset}$ *for subsets of the intersection of the challenge sets there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l} b \xleftarrow{R} \{0,1\},\ \mathsf{pp} \xleftarrow{R} \mathsf{Setup}(1^\kappa, 1^t), \\ (S_0, S_1, \mathsf{st}) \xleftarrow{R} \mathcal{A}(\mathsf{pp}), \\ (C, O) \xleftarrow{R} \mathsf{Commit}(\mathsf{pp}, S_b), \\ b^* \xleftarrow{R} \mathcal{A}^{\mathsf{OpenSubset}(\mathsf{pp}, C, O, \cdot \cap (S_0 \cap S_1))}(\mathsf{st}, C) \end{array} : \quad b^* = b \right] - \frac{1}{2} \leq \epsilon(\kappa).$$

In the *perfectly hiding* case, unbounded adversaries are being considered and $\epsilon \equiv 0$.

6.2 The Construction

In Scheme 8, we now give a construction of a set-commitment scheme. For the sake of compact representation, for a set $S \subset \mathbb{Z}_p$ we let $f_S(X) \coloneqq \prod_{s \in S}(X - s) = \sum_{i=0}^{|S|} f_i \cdot X^i$. For a group generator P, since $f_S(a)P = \sum_{i=0}^{|S|}(f_i \cdot a^i)P$, one can efficiently compute $f_S(a)P$ when given $(a^i P)_{i=0}^{|S|}$ but not a itself.

We have augmented the scheme from [HS14] by a special opening (of the form $(1, a, S)$) for the case that a set S contains the trapdoor a. (Under the t-co-DL assumption from Definition 2.12, such sets are infeasible to find.) This makes the scheme perfectly correct and perfectly hiding while still maintaining computational binding and subset-soundness.

Remark 6.6. We have defined the scheme in a way that reduces the computational complexity of the prover in the ABC system in Section 7.2.4. To improve the performance of $\mathsf{VerifySubset}$, one could define a scheme with $C \in \mathbb{G}_1$ and $W \in \mathbb{G}_2$ (for which $\mathsf{VerifySubset}$ would have to compute $f_T(a)P$).

6.2.1 Security

We prove Scheme 8 secure under the q-co-DL assumption and the generalized q-co-SDH assumption. We use both assumptions in a static way, as $q \leftarrow t$ is a system parameter and fixed a priori (i.e., it does not depend on the behavior of the adversary).

Theorem 6.7. *Scheme 8 is correct.*

Proof. Let $t, \kappa > 0$ and $(\mathsf{BG}, (a^i P, a^i \hat{P})_{i \in [t]}) \xleftarrow{R} \mathsf{Setup}(1^\kappa, 1^t)$ with $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$, let $S \subset \mathbb{Z}_p$ with $0 < |S| \leq t$ and let $\emptyset \neq T \subseteq S$. We consider two cases.

(1) $a \in S$: $\mathsf{Commit}(\mathsf{pp}, \mathsf{S})$ returns (C, O) with $C \in \mathbb{G}_1^*$ and $O = (1, a, S)$. Open on input $(C, (1, a, S))$ returns S, which shows the first property. $\mathsf{OpenSubset}$ on

Scheme 8 A set-commitment scheme.

$\mathsf{Setup}(1^\kappa, 1^t)$: On input a security parameter 1^κ and a maximum set cardinality 1^t run $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P}) \xleftarrow{R} \mathsf{BGGen}(1^\kappa)$, pick $a \xleftarrow{R} \mathbb{Z}_p$ and output $\mathsf{pp} \leftarrow (\mathsf{BG}, (a^i P, a^i \hat{P})_{i\in[t]})$, which defines message space $\mathcal{S}_{\mathsf{pp}} = \{S \subseteq \mathbb{Z}_p : 0 < |S| \leq t\}$.

$\mathsf{Commit}(\mathsf{pp}, S)$: On input $\mathsf{pp} = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i\in[t]})$ and a non-empty set $S \subset \mathbb{Z}_p$ with $|S| \leq t$:

- If for some $a' \in S$: $a'P = aP$, output $C \xleftarrow{R} \mathbb{G}_1^*$ and opening $O \leftarrow (1, a', S)$.
- Else pick $\rho \xleftarrow{R} \mathbb{Z}_p^*$, compute $C \leftarrow \rho \cdot f_S(a)P \in \mathbb{G}_1^*$ and output (C, O) with $O \leftarrow (0, \rho, S)$.

$\mathsf{Open}(\mathsf{pp}, C, O)$: On input $\mathsf{pp} = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i\in[t]})$, a commitment C and opening $O = (b, \rho, S)$: if $C \notin \mathbb{G}_1^*$ or $\rho \notin \mathbb{Z}_p^*$ or $S \not\subseteq \mathbb{Z}_p$ or $S = \emptyset$ or $|S| > t$ then return $\perp$.

- If $O = (1, a', S)$ and $a'P = aP$ then return S. Else return $\perp$.
- If $O = (0, \rho, S)$ and $C = \rho \cdot f_S(a)P$, return S. Else return $\perp$.

$\mathsf{OpenSubset}(\mathsf{pp}, C, O, T)$: On input $\mathsf{pp} = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i\in[t]})$, a commitment C, opening O and a set T, let $S \leftarrow \mathsf{Open}(\mathsf{pp}, C, O)$. If $S = \perp$, $T \not\subseteq S$ or $T = \emptyset$ then output $\perp$.

- If $O = (1, a', S)$: If $a' \in T$, return $W \leftarrow \perp$; else return $W \leftarrow f_T(a')^{-1} \cdot C$.
- If $O = (0, \rho, S)$, output $W \leftarrow \rho \cdot f_{S\setminus T}(a)P$.

$\mathsf{VerifySubset}(\mathsf{pp}, C, T, W)$: On input $\mathsf{pp} = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i\in[t]})$, a commitment C, a set T and a witness W: if $C \notin \mathbb{G}_1^*$ or $T \not\subset \mathbb{Z}_p$ or $T = \emptyset$ or $|T| > t$, return 0.

- If for some $a' \in T$: $a'P = aP$ then: If $W = \perp$, return 1; else return 0.
- Else: If $W \in \mathbb{G}_1^*$ and $e(W, f_T(a)\hat{P}) = e(C, \hat{P})$, return 1; else return 0.

input (pp, C, O, T) returns $W \leftarrow \perp$ if $a \in T$ and $W \leftarrow f_T(a)^{-1} \cdot C$ if $a \notin T$. If $a \in T$ then $\mathsf{VerifySubset}(\mathsf{pp}, C, T, W)$ returns 1 if $W = \perp$. If $a \notin T$, it returns 1 if $C, W \in \mathbb{G}_1^*$ and $e(W, f_T(a)\hat{P}) = e(C, \hat{P})$; this is satisfied, since $W \in \mathbb{G}_1^*$ and $e(W, f_T(a)\hat{P}) = e(f_T(a)^{-1} \cdot C, f_T(a)\hat{P}) = e(C, \hat{P})$.

(2) $a \notin S$: $\mathsf{Commit}(\mathsf{pp}, \mathsf{S})$ returns (C, O) with $C = \rho \cdot f_S(a)P$ and $O = (0, \rho, S)$ with $\rho \in \mathbb{Z}_p^*$. For O of this form, Open returns S, since the four clauses in its definition are satisfied. $\mathsf{OpenSubset}(\mathsf{pp}, C, O, T)$ returns $W \leftarrow \rho \cdot f_{S \setminus T}(a)P$. On input (pp, C, T, W), $\mathsf{VerifySubset}$ returns 1 if $C, W \in \mathbb{G}_1^*$ and $e(W, f_T(a)\hat{P}) = e(C, \hat{P})$. Since $\rho \in \mathbb{Z}_p^*$, $a \notin S$ we have $W = \rho \cdot f_{S \setminus T}(a)P \in \mathbb{G}_1^*$; moreover, $e(W, f_T(a)\hat{P}) = e(\rho \cdot f_S(a) \cdot f_T(a)^{-1} \cdot P, f_T(a)\hat{P}) = e(\rho \cdot f_S(a)P, \hat{P}) = e(C, \hat{P})$; so $\mathsf{VerifySubset}$ returns 1. □

Theorem 6.8. *If the t-co-DL assumption holds, then Scheme 8 is binding.*

Proof. We show that if $\mathcal{A}$ is able to output a commitment C and two valid openings to distinct sets S, S' then we can construct an adversary $\mathcal{B}$ that breaks t-co-DL: $\mathcal{B}$ obtains an instance $I = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i \in [t]})$, sets $\mathsf{pp} \leftarrow I$ and runs $\mathcal{A}(\mathsf{pp})$. If $\mathcal{A}$ outputs a collision (C, O, O'), then by $\perp \neq S \leftarrow \mathsf{Open}(\mathsf{pp}, C, O)$ and $\perp \neq S' \leftarrow \mathsf{Open}(\mathsf{pp}, C, O')$ with $S \neq S'$, it holds that $C \in \mathbb{G}_1^*$. If $O = (1, a', S)$ or $O' = (1, a', S')$ then $\mathcal{B}$ outputs a' as solution to the t-co-DL problem. Else, we have $O = (0, \rho, S), O' = (0, \rho', S')$ with $\emptyset \neq S, S' \subset \mathbb{Z}_p$, $\rho, \rho' \in \mathbb{Z}_p^*$ and:

$$\rho \cdot f_S(a)P = C = \rho' \cdot f_{S'}(a)P,$$

from which we have $\rho \cdot f_S(a) - \rho' \cdot f_{S'}(a) = 0$. Since S and S' are both non-empty and distinct, we have $\deg f_S > 0$ and $\deg f_{S'} > 0$ and $f_S \neq f_{S'}$. Furthermore, f_S and $f_{S'}$ are monic and $\rho, \rho' \neq 0$, thus $t(X) \leftarrow \rho \cdot f_S(X) - \rho' \cdot f_{S'}(X) \neq 0$ while $t(a) = 0$. Therefore, a is a root of the non-zero polynomial $t(X) \in \mathbb{Z}_p[X]$ and factoring $t(X)$ yields a. Using pp, $\mathcal{B}$ can efficiently obtain and output a as solution to the t-co-DL problem. □

Theorem 6.9. *If the generalized t-co-SDH assumption holds, then Scheme 8 is subset-sound.*

Proof. We show that if $\mathcal{A}$ is able to output (C, O, T, W), such that O is a valid opening of C to set S, $T \nsubseteq S$ and $\mathsf{VerifySubset}(\mathsf{pp}, C, T, W) = 1$, then we can construct an adversary $\mathcal{B}$ against the generalized t-co-SDH as follows. On input an instance $I = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i \in [t]})$, $\mathcal{B}$ sets $\mathsf{pp} \leftarrow I$ and runs $\mathcal{A}(\mathsf{pp})$; assume $\mathcal{A}$ breaks subset-soundness by outputting (C, O, T, W).

We first deal with the case $f_T(a) = 0$. Since $T \neq \emptyset$, and thus $f_T(X)$ is a non-constant polynomial with root a, $\mathcal{B}$ can efficiently obtain a by factoring $f_T(X)$. It then chooses $c \in \mathbb{Z}_p \setminus \{-a\}$, and outputs a solution $(1, X + c, \frac{1}{a+c}P)$ to generalized t-co-SDH.

For the rest of the proof, assume $f_T(a) \neq 0$. If $\mathcal{A}$ is successful, by $\perp \neq S \leftarrow \mathsf{Open}(\mathsf{pp}, C, O)$ and $O = (1, a', S)$ then $\mathcal{B}$ chooses $c \in \mathbb{Z}_p \setminus \{-a'\}$, and outputs a

solution $(1, X + c, \frac{1}{a'+c}P)$ to generalized t-co-SDH. Else, we have $O = (0, \rho, S)$ with $\emptyset \neq S \subset \mathbb{Z}_p$, $\rho \in \mathbb{Z}_p^*$ and

$$C = \rho \cdot f_S(a)P \in \mathbb{G}_1^*. \tag{6.1}$$

Moreover, from $\mathsf{VerifySubset}(\mathsf{pp}, C, T, W) = 1$ we have $\emptyset \neq T \subset \mathbb{Z}_p$, $|T| \leq t$, $W \in \mathbb{G}_1^*$ and $e(W, f_T(a)\hat{P}) = e(C, \hat{P})$, which by (6.1) equals $e(\rho \cdot f_S(a)P, \hat{P})$. Since $\rho \neq 0$, we have

$$e(\rho^{-1}W, f_T(a)\hat{P}) = e(f_S(a)P, \hat{P}). \tag{6.2}$$

We further distinguish two cases:

(1) $0 < |S| < |T|$. Then $0 < \deg f_S < \deg f_T \leq t$, which together with (6.2) means that $(f_S, f_T, \rho^{-1}W)$ is a solution to the generalized t-co-SDH assumption.

(2) $0 < |T| \leq |S|$. Then $0 < \deg f_T \leq \deg f_S$. Since $T \not\subseteq S$, by polynomial division we obtain h, r with $f_S(X) = h(X)f_T(X) + r(X)$ and $0 \leq \deg r < \deg f_T$. Plugging this into (6.2), we get:

$$e(\rho^{-1}W, f_T(a)\hat{P}) = e(h(a)f_T(a)P + r(a)P, \hat{P}) = e\big(h(a)P + \tfrac{r(a)}{f_T(a)}P, f_T(a)\hat{P}\big),$$

since $f_T(a) \neq 0$. This can be rewritten as

$$e(\rho^{-1}W - h(a)P, f_T(a)\hat{P}) = e\big(\tfrac{r(a)}{f_T(a)}P, f_T(a)\hat{P}\big) = e(r(a)P, \hat{P}).$$

Together with $0 \leq \deg r < \deg f_T \leq t$, this means that $(r, f_T, \rho^{-1}W - h(a)P)$ is a solution to the generalized t-co-SDH assumption, which $\mathcal{B}$ can efficiently compute. □

Theorem 6.10. *Scheme 8 is perfectly hiding.*

Proof. We consider the view of an unbounded adversary $\mathcal{A}$ in the hiding experiment and assume w.l.o.g. that every query T to the $\mathsf{OpenSubset}$ oracle satisfies $T \subset \mathbb{Z}_p$ and $\emptyset \neq T \subseteq (S_0 \cap S_1)$. We distinguish several cases.

(1) $\mathcal{A}$ chooses S_0, S_1 with $a \in S_0 \cap S_1$. Then for both $b = 0, 1$, C_b is uniformly random in $\mathbb{G}_1^*$ ($C_b \in_R \mathbb{G}_1^*$) and the jth query T_j to $\mathsf{OpenSubset}$ is answered with $\perp$ if $a \in T_j$, and with $W_{j,b} = f_T(a)^{-1} \cdot C_b$ if $a \notin T_j$. The bit b is thus information-theoretically hidden from $\mathcal{A}$.

(2) a is contained in one of the sets S_0, S_1; say $a \in S_0$. Note that for all queries T_j, we have $a \notin T_j$. If $b = 0$ then $\mathcal{A}$ receives a uniformly random C_0 and when it queries T_j to the $\mathsf{OpenSubset}$ oracle, it receives $W_{j,0} = f_{T_j}(a)^{-1} \cdot C_0$. If $b = 1$ then $\mathcal{A}$ receives $C_1 = \rho \cdot f_S(a)P$ for $\rho \in_R \mathbb{Z}_p^*$, and query T_j to the $\mathsf{OpenSubset}$ oracle returns witness $W_{j,1} = \rho \cdot f_{S \setminus T_j}(a) \cdot P = \rho \cdot f_S(a) \cdot f_{T_j}(a)^{-1} \cdot P = f_{T_j}(a)^{-1} \cdot C_1$. Hence, for both $b = 0, 1$ we have $C_b \in_R \mathbb{G}_1^*$ and $W_{j,b} = f_{T_j}(a)^{-1} \cdot C_b$ for all j; the bit b is thus information-theoretically hidden from $\mathcal{A}$.

(3) $\mathcal{A}$ chooses S_0, S_1 with $a \notin S_0 \cup S_1$. Then for $b = 0, 1$: $C_b = \rho \cdot f_{S_b}(a)P$ for $\rho \in_R \mathbb{Z}_p^*$ and a query for T_j is answered by $W_{j,b} = \rho \cdot f_{S_b \setminus T_j}(a)P = f_{T_j}(a)^{-1} \cdot C_b$.

Again for both $b = 0$ and $b = 1$, $\mathcal{A}$ receives a uniform random element C_b and query replies that do not depend on b; the bit b is thus information-theoretically hidden from $\mathcal{A}$. □

7

Attribute-Based Credentials

The question isn't 'What do we want to know about people?'
It's 'What do people want to tell about themselves?'
— Mark Zuckerberg

Anonymous credentials provide means for anonymous authentication. In particular, a credential system is a multi-party protocol involving a user, an organization (or issuer) and a verifying party. Thereby, the user can obtain a credential (on multiple attributes; in case of attribute-based credentials (ABCs)) from an organization and present the credential to some verifying party later on. While not learning any information about the user performing (a predetermined or arbitrary number of) credential showings *(anonymity)*, the verifier can still be sure that presented information (e.g., the shown attributes; in case of ABCs) is authentic *(unforgeability)*.

We distinguish two major lines of credential systems: *one-show* and *multi-show credential systems*.

The former type is typically built from blind signatures. In this case, a user obtains a blind signature from an issuer on (commitments to) attributes and, later, shows the signature, provides the shown attributes (or proves relations about them) and proves knowledge of all unrevealed attributes [Bra00, BL13a, FHS15a]. The drawback of such a blind-signature approach is that such credentials can only be shown once in an unlinkable fashion *(one-show)*. This does not necessarily mean that the verifier is able to identify a user behind a showing but that she is at least able to trace users across multiple showings—a feature that can, depending on the scenario, be both desirable and undesirable. While it can be helpful to detect, e.g., unauthorized double-usage of resources, it can

be that for some applications (like annual tickets in public transport), however, it guarantees too little privacy.

This leads us to multi-show credentials which are anonymous credential systems supporting an arbitrary number of unlinkable showings. More precisely, in a multi-show credential system, a user obtains a credential from an organization typically in a non-anonymous way but can later perform an arbitrary number of unlinkable showings. In this sense, multi-show credentials are dual to one-show credentials. They can be built in a similar vein using different types of signatures: A user obtains a signature on (commitments to) attributes, then *randomizes* the signature (such that the resulting signature is unlinkable to the issued one) and proves in ZK the correspondence of this signature with the shown attributes as well as the undisclosed attributes [CL03, CL04].

In this chapter we will give new ways to build efficient one-show and multi-show ABCs in the standard model. In Section 7.1, we will show how to build one-show ABCs in the vein of Brands from our blind-signature scheme on message vectors (cf. Scheme 5). It is the first such scheme that is based on a blind-signature scheme having security proofs in the standard model (and, moreover, being blind against malicious issuer keys). In Section 7.2, we will present a new and surprisingly efficient construction paradigm for multi-show ABC systems. It is based on SPS-EQ and set commitments and the first multi-show ABC scheme having constant-size credentials and constant communication effort during showings (i.e., independently of the number of shown attributes). Moreover, it is the first scheme that is anonymous against malicious organizations in the standard model. Last but not least, we introduce a comprehensive game-based security model for multi-show ABCs, in which we prove our construction secure.

The results in this chapter are joint work with Daniel Slamanig and Georg Fuchsbauer. This chapter details several different ABC-system constructions: The one-show credential system from [FHS15a] and the multi-show credential system from [HS14, FHS16].

7.1 Attribute-Based One-Show Credentials (aka Anonymous Credentials Light)

One-show credential systems are typically built from blind signatures following the approach from Brands [Bra00], which has been implemented in Microsoft's U-Prove [BP10]. Thereby, blind signatures ensure that no party is able to link the credential issuance to any of its showings, while different showings of the same credential are linkable. In 2013, Baldimtsi et al. [BL13b] showed that with currently known proof techniques the underlying blind-signature scheme by Brands [Bra00] cannot be proven secure. To get around this problem, they propose a generic construction of one-show credentials (in the fashion of Brands; called "Anonymous Credentials Light") secure in the ROM [BL13a]. Their credential system is based on a blind-signature scheme that they term blind signatures with attributes, for which they also give a construction based on a

non-round-optimal blind-signature scheme by Abe [Abe01].

In the following, we adapt the approach from [BL13a] and show how to build a one-show credential system from our round-optimal blind signature with attributes construction in Scheme 5, whose security proofs hold in the standard-model (in contrast to [BL13a]).

The intuition behind our construction is comparable to [BL13a], which works in the following fashion. In the *registration phase*, a user registers (once) a generalized Pedersen commitment C to her attributes and gives a ZKP of the opening (some attributes may be opened and some may remain concealed). In the *preparation* and *validation phase*, the user engages in a blind-signature-with-attributes protocol for some message m (which is considered the credential serial number) and another commitment C'. C' is a so-called combined commitment obtained from C and a second credential-specific commitment provided by the user. Finally, the credential is the user output of a blind-signature-with-attributes protocol resulting in a signature on message m and a so-called blinded Pedersen commitment C''. The latter contains the same attributes as C, but is unlinkable to C and C'. Showing a credential amounts to presenting C'' along with the blind signature and proving in ZK a desired relation about the attributes within C''.

Our construction combines Scheme 5 with efficient ZKPoKs and is conceptually simpler than the one in [BL13a]. For issuing, the user sends the issuer a blinded version $M \leftarrow (sC, sP)$ of a commitment C to the user's attributes (M corresponds to the blinded generalized Pedersen commitment in [BL13a]). In addition, the user engages in a ZKPoK (denoted PoK) proving knowledge of an opening of C (potentially revealing some of the committed attributes). The user obtains a BSV-signature (cf. Scheme 5) π on M and turns it into a blind signature σ for commitment C by running $((C,P),\sigma) \xleftarrow{R} \mathsf{ChgRep}_{\mathcal{R}}(M, \pi, \frac{1}{s}, \mathsf{pk})$. The credential consists of C, σ and the randomness r used to produce the commitment. It is showed by sending C and σ and proving in ZK a desired relation about attributes within C.

For ease of presentation, we only consider selective attribute disclosure below. We note that proofs for a rich class of relations [CDS94, CM99, BS02b] with respect to generalized Pedersen commitments, as used by our scheme, could be used instead. Henceforth, we denote by S the index set of attributes to be shown and by U those to be withheld. During a showing, a ZKPoK for a commitment $C = \sum_{i\in[n]} m_i P_i + rQ$ to attributes $\vec{m} = (m_i)_{i\in[n]} \in \mathbb{Z}_p^n$ amounts to proving

$$\mathsf{PoK}_{\mathsf{P}}\big\{\big((\alpha_j)_{j\in U}, \beta\big) \;:\; C = \textstyle\sum_{i\in S} m_i P_i + \sum_{j\in U} \alpha_j P_j + \beta Q\big\}. \tag{7.1}$$

The proof for a *blinded* commitment $(A, B) = (sC, sP)$ during the obtain phase is done as follows.

$$\mathsf{PoK}_{\mathsf{BP}}\left\{\big((\alpha_j)_{j\in U}, \beta, \gamma\big) : \begin{array}{l} A = \sum_{i\in S} m_i H_i + \sum_{j\in U} \alpha_j H_j + \beta H_Q \;\wedge \\ \bigwedge_{i\in[n]} (H_i = \gamma P_i) \wedge H_Q = \gamma Q \wedge B = \gamma P \end{array}\right\}. \tag{7.2}$$

Here the representation is with respect to bases $H_i = sP_i$, $H_Q = sQ$, which are

published and guaranteed to be correctly formed by $\mathsf{PoK}_{\mathsf{BP}}$.[1]

7.1.1 Construction

As we combine Scheme 5 with ZKPoKs, we need the following conceptual modifications. The signature $\tau = (\sigma, R, T)$ reduces to $\tau = \sigma$, since the user provides a ZKPoK proving knowledge of the randomness r in C. Moreover, verification takes C instead of $\vec{m}$ as verifiers have only access to the commitment. Consequently, algorithm Verify of Scheme 5 only runs $\mathsf{Verify}_{\mathcal{R}}$.

Setup. The issuer runs $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa, 1^n)$, where n is the number of attributes in the system, and publishes pk as her public key.

Issuing. A user with attributes $\vec{m} \in \mathbb{Z}_p^n$ runs $(\mathsf{st}, M) \xleftarrow{R} \mathcal{U}^{(1)}(\vec{m}, \mathsf{pk}; (s, r))$ (where (s, r) is the chosen randomness), sends the blinded commitment $M = (sC, sP)$ to the issuer and gives a proof $\mathsf{PoK}_{\mathsf{BP}}$ from (7.2) that M commits to $\vec{m}$ (where the sets U and S depend on the application). On success, the issuer then returns $\pi \xleftarrow{R} \mathcal{S}(M, \mathsf{sk})$ and after running $\sigma \xleftarrow{R} \mathcal{U}^{(2)}(\mathsf{st}, \pi)$ (the outputs rP and rQ are not needed), the user holds a credential (C, σ, r).

Showing. Assume a user with credential (C, σ, r) to a vector of attributes $\vec{m} \in \mathbb{Z}_p^n$ wants to conduct a selective showing of attributes with a verifier who holds the issuer's public key pk. They engage in a proof $\mathsf{PoK}_{\mathsf{P}}$ from (7.1) and the verifier additionally checks the signature for the credential by running $\mathsf{Verify}(C, \sigma, \mathsf{pk})$. If both verifications succeed, the verifier accepts the showing.

Security

Let us finally note that there is no formal security model for one-show credentials. Theorem 2 in [BL13a] informally states that a secure commitment scheme together with a blind-signature scheme with attributes implies a one-show credential system. Using the same argumentation as [BL13a], our construction yields a one-show credential system in the standard model.

7.2 Multi-Show Credentials from SPS-EQ and Set Commitments

In this section, we present an application of SPS-EQ and our set-commitment scheme from Chapter 6 by using them as basic building blocks for an ABC system.

[1] In the blindness game, given $B = sP$ from a DDH instance, these bases are simulated as $H_j \leftarrow p_j B$ and $H_Q \leftarrow qB$. We can even prove security in the malicious-signer model by extending Assumption 4.11: In addition to Q the adversary outputs $(P_i)_{i\in[n]}$ and receives $(sP_i)_{i\in[n]}$ and sQ.

Multi-show ABCs can be constructed in the following way: A user obtains a signature on (commitments to) attributes, then *randomizes* the signature (such that the resulting signature is unlinkable to the issued one) and proves in ZK the correspondence of this signature with the shown attributes as well as the undisclosed attributes [CL03, CL04]. Our approach also achieves multi-show ABCs, but differs from the latter significantly: We randomize both the signature and the message (which is a set commitment to attributes). Moreover, we use subset-opening of set commitments to enable selective constant-size showings of attributes. Thus, we do not require costly ZKPoKs over the attributes at all (which are, otherwise, at least linear in the number of shown/encoded attributes). Moreover, our ABC is the first to achieve anonymity in the malicious key model without a CRS.

We start with a discussion of the functionality and security of ABCs in Sections 7.2.1 and 7.2.2. After providing some intuition for our construction (Section 7.2.3), we present the scheme (Section 7.2.4) and discuss its security in Section 7.2.5. In Section 7.2.6, we will sketch a scheme variant that is concurrently secure in the CRS model. Finally, we give a performance and functionality comparison with other existing approaches in Section 7.2.7.

7.2.1 Model of Multi-Show ABCs

In an ABC system there are different organizations issuing credentials to different users. Users can then anonymously demonstrate possession of these credentials to verifiers. Such a system is called multi-show ABC system when transactions (issuing and showings) carried out by the same user cannot be linked. A credential cred for user i is issued by an organization for a set of attributes $\mathtt{A}$.

Definition 7.1 (Multi-show anonymous attribute-based credential (ABC) system)**.** *A* multi-show anonymous ABC system ABC *consists of the following PPT algorithms:*

$\mathsf{OrgKeyGen}(1^\kappa, 1^t)$: *A probabilistic algorithm that gets (unary representations of) a security parameter κ and an upper bound t for the size of attribute sets. It outputs a key pair* $(\mathsf{osk}, \mathsf{opk})$ *for an organization.*

$\mathsf{UserKeyGen}(1^\kappa)$: *A probabilistic algorithm that gets (the unary representation of) a security parameter κ and outputs a key pair* $(\mathsf{usk}, \mathsf{upk})$ *for a user.*

$(\mathsf{Obtain}(\mathsf{usk}, \mathsf{opk}, \mathtt{A}), \mathsf{Issue}(\mathsf{upk}, \mathsf{osk}, \mathtt{A}))$: *These algorithms are run by a user and an organization, who interact during execution.* Obtain *is a probabilistic algorithm that takes input the user's secret key* usk*, an organization's public key* opk *and a non-empty attribute set* $\mathtt{A}$ *of size* $|\mathtt{A}| \leq t$*.* Issue *is a probabilistic algorithm that takes input a user's public key* upk*, the organization's secret key* osk *and a non-empty attribute set* $\mathtt{A}$ *of size* $|\mathtt{A}| \leq t$*. At the end of this protocol,* Obtain *outputs a credential* cred *for the user for attributes* $\mathtt{A}$ *or* $\perp$ *if the interaction was not successful.*

$(\mathsf{Show}(\mathsf{opk}, \mathtt{A}, \mathtt{A}', \mathsf{cred}), \mathsf{Verify}(\mathsf{opk}, \mathtt{A}'))$: *These algorithms are run by a user and a verifier, who interact during execution.* Show *is a probabilistic algorithm that takes input the organization's public key* opk*, an attribute set* $\mathtt{A}$ *of size* $|\mathtt{A}| \leq t$*, a non-empty set* $\mathtt{A}' \subseteq \mathtt{A}$ *(representing the attributes to be shown) and a credential* cred. Verify *is a deterministic algorithm that takes input the organization's public key* opk *and a set* $\mathtt{A}'$*. At the end of the protocol,* Verify *outputs* 1 *or* 0 *indicating whether or not it accepts the credential showing.*

7.2.2 Security of ABCs

We now present an appropriate security model for multi-show ABCs, which is a game-based model in the vein of group signatures [BSZ05] and considers malicious organization keys. We note that there are no other comprehensive models available for ABC systems—apart from independently developed very strong simulation-based notions in [CKL+14, CDHK15].

Overview

We start with a high-level overview of the required security properties and note that we consider only a single organization in our model of unforgeability and anonymity (since all organizations have independent signing keys, the extension to multiple organizations is straightforward):

Correctness: A showing of a credential with respect to a non-empty set $\mathtt{A}'$ of attributes and values must always verify if the credential was issued honestly with respect to some $\mathtt{A}$ with $\mathtt{A}' \subseteq \mathtt{A}$.

Unforgeability: A user cannot perform a valid showing of attributes for which she does not possess a credential. Moreover, no coalition of malicious users can combine their credentials and prove possession of a set of attributes which no single member has. This holds even after seeing showings of arbitrary credentials by honest users.

Anonymity: During a showing, no verifier and no (malicious) organization (even if they collude) should be able to identify the user or find out anything about the user, except for the fact that she owns a valid credential for the shown attributes. Furthermore, different showings of a user using the same credential are unlinkable.

Definitions

In the following, we provide formal definitions of these properties, for which we introduce several global variables and oracles.

Global variables. At the beginning of each experiment, either the experiment computes an organization key pair $(\mathsf{osk}, \mathsf{opk})$ or the adversary outputs opk. In the anonymity game there is a bit b, which the adversary must guess.

In order to keep track of all the users, in particular all honest and corrupt users and those whose secret keys and credentials have leaked, we introduce the sets $\mathtt{U}$, $\mathtt{HU}$, $\mathtt{CU}$ and $\mathtt{KU}$, respectively. We use the lists $\mathtt{UPK}$, $\mathtt{USK}$, $\mathtt{CRED}$, $\mathtt{ATTR}$ and $\mathtt{I2U}$ to track issued user public and secret keys, credentials and corresponding attributes and to which user they were issued. Furthermore, we use the sets J_{LoR} and I_{LoR} to store the issuance indices and corresponding users that have been set during the first call to the left-or-right oracle in the anonymity game.

Oracles. The oracles are as follows:

$\mathcal{O}^{\mathtt{HU}+}(i)$: It takes input a user identity i. If $i \in \mathtt{U}$, it returns $\bot$. Otherwise, it creates a new honest user i by running $(\mathtt{USK}[i], \mathtt{UPK}[i]) \xleftarrow{R} \mathsf{UserKeyGen}(1^\kappa)$, adding i to $\mathtt{U}$ and to $\mathtt{HU}$ and returning $\mathtt{UPK}[i]$.

$\mathcal{O}^{\mathtt{CU}+}(i, \mathsf{upk})$: It takes input a user identity i and a user public key upk. If $i \in \mathtt{U}$, it returns $\bot$. Otherwise, it adds user i to the sets $\mathtt{U}$ and $\mathtt{CU}$, and sets $\mathtt{UPK}[i] \leftarrow \mathsf{upk}$.

$\mathcal{O}^{\mathtt{KU}+}(i)$: It takes input a user identity i. If $i \notin \mathtt{HU}$ or $i \in I_{LoR}$, it returns $\bot$. Otherwise, it reveals the secret key and all credentials of user i by returning $\mathtt{USK}[i]$ and $\mathtt{CRED}[j]$ for all j with $\mathtt{I2U}[j] = i$. It removes i from $\mathtt{HU}$ and adds it to $\mathtt{KU}$.

$\mathcal{O}^{\mathsf{ObtIss}}(i, \mathtt{A})$: It takes input a user identity i and a set of attributes $\mathtt{A}$. If $i \notin \mathtt{HU}$, it returns $\bot$. Otherwise, it runs

$$(\mathsf{cred}, \top) \xleftarrow{R} (\mathsf{Obtain}(\mathtt{USK}[i], \mathsf{opk}, \mathtt{A}), \mathsf{Issue}(\mathtt{UPK}[i], \mathsf{osk}, \mathtt{A})).$$

If $\mathsf{cred} = \bot$, it returns $\bot$. Else, it appends $(i, \mathsf{cred}, \mathtt{A})$ to $(\mathtt{I2U}, \mathtt{CRED}, \mathtt{ATTR})$ and returns $\top$.

$\mathcal{O}^{\mathsf{Obtain}}(i, \mathtt{A})$: It takes input a user identity i and a set of attributes $\mathtt{A}$. If $i \notin \mathtt{HU}$, it returns $\bot$. Otherwise, it runs

$$(\mathsf{cred}, \cdot) \xleftarrow{R} (\mathsf{Obtain}(\mathtt{USK}[i], \mathsf{opk}, \mathtt{A}), \cdot),$$

where the Issue part is executed by the caller (the dishonest organization). If $\mathsf{cred} = \bot$, it returns $\bot$. Else, it appends $(i, \mathsf{cred}, \mathtt{A})$ to $(\mathtt{I2U}, \mathtt{CRED}, \mathtt{ATTR})$ and returns $\top$.

$\mathcal{O}^{\mathsf{Issue}}(i, \mathtt{A})$: It takes input a user identity i and a set of attributes $\mathtt{A}$. If $i \notin \mathtt{CU}$, it returns $\bot$. Otherwise, it runs

$$(\cdot, I) \xleftarrow{R} (\cdot, \mathsf{Issue}(\mathtt{UPK}[i], \mathsf{osk}, \mathtt{A})),$$

where the Obtain part is executed by the caller (the dishonest user). If $I = \bot$, it returns $\bot$. Else, it appends $(i, \bot, \mathtt{A})$ to $(\mathtt{I2U}, \mathtt{CRED}, \mathtt{ATTR})$ and returns $\top$.

$\mathcal{O}^{\mathsf{Show}}(j, \mathtt{A}')$: It takes input an index of an issuance j and a set of attributes $\mathtt{A}'$. Let $i \leftarrow \mathtt{I2U}[j]$. If $i \notin \mathtt{HU}$, it returns $\perp$. Otherwise, it runs

$$(S, \cdot) \xleftarrow{R} (\mathsf{Show}(\mathsf{opk}, \mathtt{ATTR}[j], \mathtt{A}', \mathtt{CRED}[j]), \cdot),$$

where the Verify part is executed by the caller (the dishonest verifier).

$\mathcal{O}^{LoR}(j_0, j_1, \mathtt{A}')$: It takes two issuance indexes j_0 and j_1 and a set of attributes $\mathtt{A}'$. If $J_{LoR} \neq \emptyset$ and $J_{LoR} \neq \{j_0, j_1\}$, it returns $\perp$. Let $i_0 \leftarrow \mathtt{I2U}[j_0]$ and $i_1 \leftarrow \mathtt{I2U}[j_1]$. If $J_{LoR} = \emptyset$, then it sets $J_{LoR} \leftarrow \{j_0, j_1\}$ and $I_{LoR} \leftarrow \{i_0, i_1\}$. If $i_0, i_1 \notin \mathtt{HU}$ or $\mathtt{A}' \not\subseteq \mathtt{ATTR}[j_0] \cap \mathtt{ATTR}[j_1]$, it returns $\perp$. Else, it runs

$$(S, \cdot) \xleftarrow{R} (\mathsf{Show}(\mathsf{opk}, \mathtt{ATTR}[j_b], \mathtt{A}', \mathtt{CRED}[j_b]), \cdot),$$

(with b set by the experiment) where the Verify part is executed by the caller.

Using the global variables and oracles just defined, we now define security of an ABC system:

Definition 7.2 (Correctness)**.** *A multi-show anonymous ABC system* ABC *is* correct *if for all $\kappa > 0$, all $t > 0$, all attribute sets $\mathtt{A}$ with $0 < |\mathtt{A}| \leq t$, all $\emptyset \neq \mathtt{A}' \subseteq \mathtt{A}$, all choices of organization key pairs $(\mathsf{osk}, \mathsf{opk}) \xleftarrow{R} \mathsf{OrgKeyGen}(1^\kappa, 1^t)$, all choices of user key pairs $(\mathsf{usk}, \mathsf{upk}) \xleftarrow{R} \mathsf{UserKeyGen}(1^\kappa)$ and all choices of $(\mathsf{cred}, \top) \xleftarrow{R} (\mathsf{Obtain}(\mathsf{usk}, \mathsf{opk}, \mathtt{A}), \mathsf{Issue}(\mathsf{upk}, \mathsf{osk}, \mathtt{A}))$ it holds that:*

$$\Pr\left[\, (\top, 1) \xleftarrow{R} (\mathsf{Show}(\mathsf{opk}, \mathtt{A}, \mathtt{A}', \mathsf{cred}), \mathsf{Verify}(\mathsf{opk}, \mathtt{A}'))\, \right] = 1.$$

Definition 7.3 (Unforgeability)**.** *A multi-show anonymous ABC system* ABC *is* unforgeable *if for all $t > 0$ and all PPT adversaries $\mathcal{A}$ having oracle access to $\mathcal{O} := \{\mathcal{O}^{\mathtt{HU}+}, \mathcal{O}^{\mathtt{CU}+}, \mathcal{O}^{\mathtt{KU}+}, \mathcal{O}^{\mathsf{ObtIss}}, \mathcal{O}^{\mathsf{Issue}}, \mathcal{O}^{\mathsf{Show}}\}$ there is a negligible function $\epsilon(\cdot)$ such that:*

$$\Pr\left[\begin{array}{ll} (\mathsf{osk}, \mathsf{opk}) \xleftarrow{R} \mathsf{OrgKeyGen}(1^\kappa, 1^t), & b^* = 1 \\ (\mathtt{A}', \mathsf{st}) \xleftarrow{R} \mathcal{A}^{\mathcal{O}}(\mathsf{opk}), & : \ \wedge\, \forall j : \mathtt{I2U}[j] \in \mathtt{KU} \cup \mathtt{CU} \\ (\cdot, b^*) \xleftarrow{R} (\mathcal{A}(\mathsf{st}), \mathsf{Verify}(\mathsf{opk}, \mathtt{A}')) & \Rightarrow \mathtt{A}' \not\subseteq \mathtt{ATTR}[j] \end{array}\right] \leq \epsilon(\kappa).$$

Definition 7.4 (Anonymity)**.** *A multi-show anonymous ABC system* ABC *is* anonymous *if for all $t > 0$ and all PPT adversaries $\mathcal{A}$ having oracle access to $\mathcal{O} := \{\mathcal{O}^{\mathtt{HU}+}, \mathcal{O}^{\mathtt{CU}+}, \mathcal{O}^{\mathtt{KU}+}, \mathcal{O}^{\mathsf{Obtain}}, \mathcal{O}^{\mathsf{Show}}, \mathcal{O}^{LoR}\}$ there is a negligible function $\epsilon(\cdot)$ such that:*

$$\Pr\left[\begin{array}{ll} b \xleftarrow{R} \{0,1\},\ (\mathsf{opk}, \mathsf{st}) \xleftarrow{R} \mathcal{A}(1^\kappa, 1^t), & : \ b^* = b \\ b^* \xleftarrow{R} \mathcal{A}^{\mathcal{O}}(\mathsf{st}) & \end{array}\right] - \frac{1}{2} \leq \epsilon(\kappa).$$

7.2.3 Intuition of Our Construction

Our construction of ABCs is based on SPS-EQ, on set commitments with subset openings and on a *single* constant-size PoK for guaranteeing freshness. In

contrast to this, the complexity of PoKs in existing ABC systems [Bra00, CL01, CL03, CL04, CL11, CL13] is linear in the number of shown (or even issued) attributes. However, aside from selective disclosure of attributes, they usually allow to prove statements about non-revealed attribute values, such as AND, OR and NOT, interval proofs, as well as conjunctions and disjunctions of the aforementioned. We achieve less expressiveness; our construction supports selective disclosure as well as AND statements about attributes (as the constructions in [CL11, CL13, CDHK15], of which only the latter also achieves constant-size showings). A user can thus either open some attributes and their corresponding values or solely prove that some attributes are encoded in the respective credential without revealing their concrete values. Note that one can always associate sets of values to attributes, so that one is not required to reveal the full attribute value, but only predefined "statements" about the attribute value, e.g., "01.01.1980","$>$ 16", or "$>$ 18" for an attribute label `birthdate`. This allows emulation of proving properties about attribute values and, thus, enhances the expressiveness of the system.

Outline

We assume attributes to be values from $\mathbb{Z}_p$ and note that we can define attributes of arbitrary format by using a collision-resistant hash function $h\colon \{0,1\}^* \to \mathbb{Z}_p$. In our construction a credential cred of user i consists of a group element C, a scalar $r \in \mathbb{Z}_p^*$, a modified opening O of C (not containing the attributes) and an SPS-EQ signature σ on $(C, r \cdot C, P)$. The element C is a set commitment to a set of attributes $\mathtt{A} \subset \mathbb{Z}_p$, whose randomness is the user secret usk (thus, its opening O contains usk or the commitment trapdoor a, if $a \in \mathtt{A}$). Using usk in that way allows us to efficiently demonstrate knowledge of the secret during issuing and is, moreover, important to achieve anonymity (omitting usk in our construction would immediately break anonymity). The values C and r define an equivalence class $[(C, r \cdot C, P)]_{\mathcal{R}}$ that is unique for each credential with overwhelming probability. The scalar r and the third credential component are merely artifacts of the unforgeability proof, i.e., to make the reduction work. During a showing, a random representative of this class, $(C_1, C_2, C_3) \xleftarrow{R} [(C, r \cdot C, P)]_{\mathcal{R}}$, together with a consistently updated signature σ' is presented. The randomized commitment C_1 is then subset-opened to the shown attributes $\mathtt{A}' \subseteq \mathtt{A}$ (representing selective disclosure). Hence, showings additionally include a witness W and a verifier checks whether the encodings of the disclosed attributes and W give a valid subset opening of C_1. In order to guarantee freshness, the prover also performs a constant-size ZKPoK of the DL of C_2 to base C_1 (which is the randomness r) and the discrete logarithm of C_3 to base P (the randomizer used for obtaining (C_1, C_2, C_3) from $(C, r \cdot C, P)$).

We now give more details on attribute representation and freshness.

Example 7.5. To give an idea of the expressiveness of our construction, we include an example of an attribute set $\mathtt{A}$. We are given a user with the following

set of attribute and value strings (which are hashed into $\mathbb{Z}_p$ via h):

$$\mathtt{A} = \{h(\text{“}\mathtt{gender},\mathtt{male}\text{”}), h(\text{“}\mathtt{birthdate},\mathtt{01.01.1980}\text{”}),\\ h(\text{“}\mathtt{drivinglicense},\#\text{”}), h(\text{“}\mathtt{drivinglicense},\mathtt{car}\text{”})\}.$$

Note that # indicates an attribute value that allows to prove the possession of the attribute without revealing any concrete value. A showing could, for instance, involve the following attributes $\mathtt{A}'$ and its hidden complement $\mathtt{A} \setminus \mathtt{A}'$:

$$\mathtt{A}' = \{h(\text{“}\mathtt{gender},\mathtt{male}\text{”}), h(\text{“}\mathtt{drivinglicense},\#\text{”})\},\\ \mathtt{A} \setminus \mathtt{A}' = \{h(\text{“}\mathtt{birthdate},\mathtt{01.01.1980}\text{”}), h(\text{“}\mathtt{drivinglicense},\mathtt{car}\text{”})\}.$$

Freshness. We have to guarantee that no valid showing transcript can be replayed by someone not in possession of the credential. To do so, we require the user to conduct a ZKPoK $\mathsf{PoK}\{\beta : C_3 = \beta P\}$ of the DL of the third component $C_3 = \mu P$ of a shown credential $\mathsf{cred}' = ((C_1, C_2, C_3), \sigma')$, i.e., the randomizer μ used in the showing protocol. This guarantees that we have a fresh challenge for every showing. For the unforgeability proof to work out, the user additionally proves knowledge of $r = \log_{C_1} C_2$ by conducting a ZKPoK $\mathsf{PoK}\{\alpha : C_2 = \alpha C_1\}$. We use the compact notation $\Pi^{\mathcal{R}_\mathsf{F}}(C_1, C_2, C_3)$ for the AND-composition of both proofs, i.e., $\Pi^{\mathcal{R}_\mathsf{F}}(C_1, C_2, C_3) := \mathsf{PoK}\{(\alpha, \beta) : C_2 = \alpha C_1 \ \wedge \ C_3 = \beta P\}$.

Malicious Organization Keys. In contrast to anonymity notions usually considered for ABCs, we aim for anonymity of constructions to hold even against adversaries that generate the organization keys maliciously. Moreover, we target the standard model. To this end, organization public keys consist of an SPS-EQ public key pk and the set-commitment parameters pp_SC. Furthermore, we augment the issuing protocol sketched above and let the (malicious) organization prove knowledge of a secret key that is consistent with its public key (which allows us to extract the signing key in the anonymity proof).

For an SPS-EQ scheme $\mathsf{SPS\text{-}EQ}$ we define an NP-relation $\mathcal{R}_\mathsf{VK}$, whose statements and witnesses are public and private keys, i.e.: $(\mathsf{pk}, \mathsf{sk}) \in \mathcal{R}'_\mathsf{VK} \iff \mathsf{VKey}_\mathcal{R}(\mathsf{sk}, \mathsf{pk}) = 1$. In our proof of anonymity we additionally need to extract the set-commitment trapdoor $a \in \mathbb{Z}_p$, so we augment the above relation to:

$$((aP, \mathsf{pk}), (w_1, w_2)) \in \mathcal{R}_\mathsf{VK} \iff (aP = w_1 P \ \wedge \ \mathsf{VKey}_\mathcal{R}(w_2, \mathsf{pk}) = 1),$$

where aP stems from the set-commitment parameters pp_SC contained in opk. For the sake of compactness, we use the notation $\Pi^{\mathcal{R}_\mathsf{VK}}(\mathsf{opk})$ and require the proof to be a perfect ZKPoK.

ZKPoKs and Concurrent Security. We will consider both ZKPoKs in a black-box way. They can be efficiently instantiated using, e.g., the 4-move ZKPoK proof systems from [CDM00], which is based on Σ-protocols and features rewindable black-box access to the verifier (Definition 2.29). We refer the reader to Section 2.6 for more details on that. Note, however, that the ZKPoKs

from [CDM00] are not concurrently secure. Hence, any instantiation of Scheme 9 with them, yields a non-concurrently secure ABC. In other words, each organization, each user and each verifier must not run more than one protocol execution at once. In Section 7.2.6, we will discuss a concurrently secure scheme variant in the CRS model.

7.2.4 The Construction of the ABC System

Our ABC system is based on any perfectly-adapting SPS-EQ $\mathsf{SPS\text{-}EQ}$ and the set-commitment scheme SC in Scheme 8 (cf. Section 6.2) and is described in Scheme 9.

In particular, since the organization public key is fully determined by the adversary (for malicious-key anonymity), we assume the bilinear-group generation algorithm of $\mathsf{SPS\text{-}EQ}$ and the one inside the set-commitment setup algorithm to be deterministic[2,3] and produce the same bilinear group for each security parameter. We will base our proofs on assumptions that are modified accordingly, i.e., that are with respect to a deterministic BGGen producing the same bilinear group for each security parameter.

Modified Set-Commitment Algorithms

For the sake of readability, we use custom variants of the set-commitment algorithms Commit and $\mathsf{OpenSubset}$ of scheme SC: Commit' and $\mathsf{OpenSubset}'$.

Commit' gives partial control over the randomness ρ used during the computation of the commitment and returns a modified opening not containing the set (as we include the opening into our credentials, this would lead to an artificial blow-up of the credential size). In particular, it returns a commitment with randomness ρ if $a \notin S$ and a uniformly random commitment otherwise.

$\mathsf{Commit}'(\mathsf{pp}, S, \rho)$: On input $\mathsf{pp} = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i\in[t]})$, a non-empty set $S \subset \mathbb{Z}_p$ with $|S| \leq t$ and a scalar $\rho \in \mathbb{Z}_p^*$:

- If for some $a' \in S$: $a'P = aP$, output $C \xleftarrow{R} \mathbb{G}_1^*$ and opening $O \leftarrow (1, a')$. *(as in* Commit *except for not including* S *into* O*)*
- Else compute $C \leftarrow \rho \cdot f_S(a)P \in \mathbb{G}_1^*$ and output (C, O) with $O \leftarrow (0, \rho)$. *(ρ was given input and is not drawn internally as in* Commit*; moreover,* S *is not included into* O*)*

We adapt $\mathsf{OpenSubset}'$ to deal with rerandomized commitments.

$\mathsf{OpenSubset}'(\mathsf{pp}, C, O, \mu, T)$: On input $\mathsf{pp} = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i\in[t]})$, a commitment C, opening O, a scalar $\mu \in \mathbb{Z}_p^*$ and a set T, let $S \leftarrow \mathsf{Open}(\mathsf{pp}, \mu^{-1} \cdot C, O)$.

[2] As already pointed out in Chapter 4, this is, e.g., the case for BN curves [BN06]; the most common choice for Type-3 pairings.

[3] Hence, the only randomness used by the set-commitment setup algorithm is the one used for picking the commitment trapdoor. Inside $\mathsf{OrgKeyGen}$, we will make this randomness explicit.

If $S = \bot$, $T \not\subseteq S$ or $T = \emptyset$ then output $\bot$. *(contrary to* OpenSubset, Open *is being run on* $\mu^{-1} \cdot C$ *instead of* C*)*

- If $O = (1, a', S)$: If $a' \in T$, return $W \leftarrow \bot$; else return $W \leftarrow f_T(a')^{-1} \cdot C$. *(as in* OpenSubset*)*
- If $O = (0, \rho, S)$, output $W \leftarrow \mu \cdot \rho \cdot f_{S \setminus T}(a)P$. *(contrary to* OpenSubset*, W gets additionally multiplied by μ)*

Optimizations

Note that the first move in the showing protocol can be combined with the first move of $\Pi^{\mathcal{R}_\mathsf{F}}$, meaning the showing protocol consists of a total of four moves, when using 4-move ZKPoKs. Also, the moves in the issue protocol can be collapsed. Furthermore, note that the issuance can be made more efficient with regard to both communication complexity and computational effort, as osk contains set-commitment trapdoor a (for the sake of presentation based on the introduced set-commitment algorithms, we, however, do not augment the scheme in that particular way).

7.2.5 Security

The correctness of Scheme 9 follows by inspection. Subsequently, we will prove the following:

Theorem 7.6. *Let* $\Pi^{\mathcal{R}_\mathsf{F}}$ *and* $\Pi^{\mathcal{R}_\mathsf{VK}}$ *be ZKPoKs. If the* t*-co-DL assumption holds,* SC *is subset-sound and* SPS-EQ *is EUF-CMA-secure, then Scheme 9 is unforgeable.*

Theorem 7.7. *Let* $\Pi^{\mathcal{R}_\mathsf{F}}$ *be a ZKPoK. If the* SPS-EQ *has a class-hiding message space and perfectly adapts signatures, then Scheme 9 is anonymous.*

Proof of Theorem 7.6

In the proof of unforgeability we distinguish whether the adversary won the game by forging a signature, breaking subset-opening soundness of the commitment scheme or computing a discrete logarithm. We can efficiently determine which was the case since the knowledge extractor of the ZKPoK $\Pi^{\mathcal{R}_\mathsf{F}}$ lets us extract the used credential.

Proof. We first introduce the following syntactic changes to the scheme and the experiment, which let us distinguish different types of forgeries: (1) We include value $R = r \cdot C$ into credentials cred output by Obtain (they are now of the form $\mathsf{cred} = ((C, R), \sigma, r, O)$). (2) When the adversary makes a valid call to $\mathcal{O}^{\mathsf{Issue}}$, the experiment receives the values C, R and produces a signature σ; instead of appending $\bot$ to the list CRED, the oracle now appends $((C, R), \sigma, \bot, \bot)$ to the list. (Altogether, this lets us check whether the adversary forged a signature when winning the unforgeability game.)

Scheme 9 A multi-show ABC system.

$\mathsf{OrgKeyGen}(1^\kappa, 1^\mathsf{t})$: Given $\kappa, t > 0$ in unary, pick $a \xleftarrow{R} \mathbb{Z}_p$, run $\mathsf{pp_{SC}} = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i\in[t]}) \leftarrow \mathsf{Setup}(1^\kappa, 1^t; a)$, run $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}_\mathcal{R}(\mathsf{BG}, 1^\ell)$ for $\ell = 3$ and return $(\mathsf{osk}, \mathsf{opk}) \leftarrow ((a, \mathsf{sk}), (\mathsf{pp_{SC}}, \mathsf{pk}))$.

$\mathsf{UserKeyGen}(1^\kappa)$: Given security parameter κ in unary, run $\mathsf{BG} \leftarrow \mathsf{BGGen}_\mathcal{R}(1^\kappa)$, pick $\mathsf{usk} \xleftarrow{R} \mathbb{Z}_p^*$, set $\mathsf{upk} \leftarrow \mathsf{usk} \cdot P$ and return $(\mathsf{usk}, \mathsf{upk})$.

$(\mathsf{Obtain}, \mathsf{Issue})$: Using $\Pi^{\mathcal{R}_{\mathsf{VK}}}(\mathsf{opk}) := \mathsf{PoK}\{(\alpha, \vec{\beta}) : \alpha P = aP \wedge \mathsf{VKey}_\mathcal{R}(\vec{\beta}, \mathsf{pk}) = 1\}$, Obtain and Issue interact as follows:

$\mathsf{Obtain}(\mathsf{usk}, \mathsf{opk}, \mathtt{A})$		$\mathsf{Issue}(\mathsf{upk}, \mathsf{osk}, \mathtt{A})$
If $\mathtt{A} = \emptyset \vee \mathtt{A} \not\subset \mathbb{Z}_p \vee \lvert\mathtt{A}\rvert > t$ return $\perp$		If $\mathtt{A} = \emptyset \vee \mathtt{A} \not\subset \mathbb{Z}_p \vee \lvert\mathtt{A}\rvert > t$ return $\perp$
$\mathsf{BG} \leftarrow \mathsf{BGGen}_\mathcal{R}(1^\kappa)$		
If $\Pi^{\mathcal{R}_{\mathsf{VK}}}$ fails, return $\perp$	$\xleftrightarrow{\Pi^{\mathcal{R}_{\mathsf{VK}}}(\mathsf{opk})}$	
$(C, O) \xleftarrow{R} \mathsf{Commit}'(\mathsf{pp_{SC}}, \mathtt{A}, \mathsf{usk})$		
$r \xleftarrow{R} \mathbb{Z}_p^*,\ R \leftarrow r \cdot C$	$\xrightarrow{C, R}$	If $e(C, \hat{P}) \neq e(\mathsf{upk}, f_\mathtt{A}(a)\hat{P})$ and $\forall a' \in \mathtt{A} : a'P \neq aP$, return $\perp$
If $\mathsf{Verify}_\mathcal{R}((C, R, P), \sigma, \mathsf{pk}) = 0$ return $\perp$	$\xleftarrow{\sigma}$	Else $\sigma \xleftarrow{R} \mathsf{Sign}_\mathcal{R}((C, R, P), \mathsf{sk})$
Return $\mathsf{cred} \leftarrow (C, \sigma, r, O)$		

$(\mathsf{Show}, \mathsf{Verify})$: Using $\Pi^{\mathcal{R}_\mathsf{F}}(C_1, C_2, C_3) := \mathsf{PoK}\{(\alpha, \beta) : C_2 = \alpha C_1 \ \wedge \ C_3 = \beta P\}$, Show and Verify interact as follows:

$\mathsf{Show}(\mathsf{opk}, \mathtt{A}, \mathtt{A}', \mathsf{cred})$		$\mathsf{Verify}(\mathsf{opk}, \mathtt{A}')$
Let $\mathsf{cred} = (C, \sigma, r, O)$; $\mu \xleftarrow{R} \mathbb{Z}_p^*$		
$\mathsf{cred}' \xleftarrow{R} \mathsf{ChgRep}_\mathcal{R}((C, r\cdot C, P), \sigma, \mu, \mathsf{pk})$		
If $\mathsf{cred}' = \perp$, return $\perp$		
Let $\mathsf{cred}' = ((C_1, C_2, C_3), \sigma')$		
$\vec{C} \leftarrow (C_1, C_2, C_3),\ O' \leftarrow (O, \mathtt{A})$		
$W \leftarrow \mathsf{OpenSubset}'(\mathsf{pp_{SC}}, C_1, O', \mu, \mathtt{A}')$	$\xrightarrow{\mathsf{cred}', W}$	
	$\xleftrightarrow{\Pi^{\mathcal{R}_\mathsf{F}}(\vec{C})}$	If $\Pi^{\mathcal{R}_\mathsf{F}}$ fails, return 0
		Return $\big(\mathsf{Verify}_\mathcal{R}(\mathsf{cred}', \mathsf{pk}) \wedge \mathsf{VerifySubset}(\mathsf{pp_{SC}}, C_1, \mathtt{A}', W)\big)$

Assume now an efficient adversary $\mathcal{A}$ wins the unforgeability game (Definition 7.3) with non-negligible probability and let $((C_1^*, C_2^*, C_3^*), \sigma^*)$ be the message-signature pair it uses and W^* be the witness for an attribute set $\mathtt{A}'^* \not\subseteq \mathtt{ATTR}[j]$, for all j with $\mathtt{I2U}[j] \in \mathtt{KU} \cup \mathtt{CU}$; moreover, the ZKPoK $\Pi^{\mathcal{R}_\mathsf{F}}(C_1^*, C_2^*, C_3^*)$ verifies. We distinguish the following cases:

Type 1: $[(C_1^*, C_2^*, C_3^*)]_\mathcal{R} \neq [(C, R, P)]_\mathcal{R}$ for $((C, R), \sigma, *, *) = \mathtt{CRED}[j]$ for all issuance indexes j (i.e., $\mathtt{I2U}[j] \in \mathtt{KU} \cup \mathtt{CU} \cup \mathtt{HU}$).

Since $((C_1^*, C_2^*, C_3^*), \sigma^*)$ is a valid pair, we are dealing with a signature forgery. Using $\mathcal{A}$ we construct an adversary $\mathcal{B}$ that breaks the EUF-CMA security of the SPS-EQ scheme.

Type 2: $[(C_1^*, C_2^*, C_3^*)]_\mathcal{R} = [(C, R, P)]_\mathcal{R}$ where $((C, R), \sigma, *, *) = \mathtt{CRED}[j]$ for some j with $\mathtt{I2U}[j] \in \mathtt{KU} \cup \mathtt{CU}$.

Since $\mathcal{A}$ only wins if $\mathtt{A}' \not\subseteq \mathtt{ATTR}[j]$, it must have broken the subset-soundness property of set commitments. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that breaks subset soundness of the set-commitment scheme SC.

Type 3: $[(C_1^*, C_2^*, C_3^*)]_\mathcal{R} = [(C, R, P)]_\mathcal{R}$ where $((C, R), \sigma, r, O) = \mathtt{CRED}[j]$ for some j with $\mathtt{I2U}[j] \in \mathtt{HU}$.

Then, we use $\mathcal{A}$ to break q-co-DLP.

Type 1. $\mathcal{B}$ interacts with the challenger $\mathcal{C}$ in the EUF-CMA game of the SPS-EQ scheme and $\mathcal{B}$ simulates the ABC-unforgeability game for $\mathcal{A}$.

$\mathcal{C}$ sets up $(\mathsf{sk}, \mathsf{pk})$ for the SPS-EQ scheme with $\ell = 3$ and gives pk to $\mathcal{B}$, which contains a bilinear-group description $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P}) = \mathsf{BGGen}_\mathcal{R}(1^\kappa)$. Then, $\mathcal{B}$ picks $a \xleftarrow{R} \mathbb{Z}_p$ and defines $\mathsf{pp}_\mathsf{sc} \leftarrow (\mathsf{BG}, (a^i P, a^i \hat{P})_{i \in [t]})$; it moreover sets $(\mathsf{osk}, \mathsf{opk}) \leftarrow ((a, \bot), (\mathsf{pp}_\mathsf{sc}, \mathsf{pk}))$. Next $\mathcal{B}$ runs $\mathcal{A}(\mathsf{opk})$ and simulates the environment and the oracles. All oracles are as in the real game, except for the following oracles, whose simulations deviate from the real game as follows:

$\mathcal{O}^{\mathsf{ObtIss}}(i, \mathtt{A})$: Instead of creating a credential by running $(\mathsf{Obtain}, \mathsf{Issue})$, $\mathcal{B}$ appends $(i, \bot, \mathtt{A})$ to $(\mathtt{I2U}, \mathtt{CRED}, \mathtt{ATTR})$.

$\mathcal{O}^{\mathtt{KU}+}(i)$: For all j such that $\mathtt{I2U}[j] = i$, $\mathcal{B}$ computes $(C, O) \xleftarrow{R} \mathsf{Commit}'(\mathsf{pp}_\mathsf{sc}, \mathtt{ATTR}[j], \mathtt{USK}[i])$, chooses $r \xleftarrow{R} \mathbb{Z}_p^*$ and queries $\mathcal{C}$'s signing oracle $\mathsf{Sign}_\mathcal{R}(\cdot, \mathsf{sk})$ on message $(C, r \cdot C, P)$ to obtain σ; $\mathcal{B}$ sets $\mathtt{CRED}[j] \leftarrow ((C, r \cdot C), \sigma, r, O)$ and runs this oracle as in the real game.

$\mathcal{O}^{\mathsf{Issue}}(i, \mathtt{A})$: $\mathcal{B}$ runs this oracle by running the simulator $\mathcal{S}$ of ZKPoK $\Pi^{\mathcal{R}_\mathsf{VK}}(\mathsf{opk})$ (as it does not know $\mathsf{sk} = \mathsf{osk}[2]$), moreover instead of signing (C, R, P), $\mathcal{B}$ obtains the signature σ from $\mathcal{C}$'s signing oracle.

$\mathcal{O}^{\mathsf{Show}}(j, \mathtt{A}')$: $\mathcal{B}$ computes $(C, O) \xleftarrow{R} \mathsf{Commit}'(\mathsf{pp}_\mathsf{sc}, \mathtt{ATTR}[j], \mathtt{USK}[\mathtt{I2U}[j]])$, chooses $r \xleftarrow{R} \mathbb{Z}_p^*$ and queries a signature σ on message $(C, r \cdot C, P)$ to $\mathcal{C}$'s signing oracle. $\mathcal{B}$ sets $\mathtt{CRED}[j] \leftarrow ((C, r \cdot C), \sigma, r, O)$ and runs the oracle as in the real game.

When $\mathcal{A}$ outputs $(\mathtt{A}'^*, \mathsf{st})$, $\mathcal{B}$ runs $\mathcal{A}(\mathsf{st})$ and interacts with $\mathcal{A}$ as verifier in a showing protocol. If $\mathcal{A}$ delivers a valid showing using $((C_1^*, C_2^*, C_3^*), \sigma^*)$ and conducting the ZKPoK $\Pi^{\mathcal{R}_\mathsf{F}}(C_1^*, C_2^*, C_3^*)$, then $\mathcal{B}$ runs the knowledge extractor $\mathcal{E}$ of $\Pi^{\mathcal{R}_\mathsf{F}}(C_1^*, C_2^*, C_3^*)$ to obtain a witness $w = (r'', \mu)$ such that $C_3^* = \mu P$. If there is a credential $\bot \neq ((C', R'), \sigma', *, *) \in \mathtt{CRED}$ such that $(C', R', P) = \mu^{-1} \cdot (C_1^*, C_2^*, C_3^*)$ then $\mathcal{B}$ aborts. (In this case, the forgery is not of Type 1.) Otherwise, $\mathcal{B}$ outputs $(M^*, \sigma^*) \xleftarrow{R} \mathsf{ChgRep}_\mathcal{R}((C_1^*, C_2^*, C_3^*), \sigma^*, \mu^{-1}, \mathsf{pk})$ as a forgery to $\mathcal{C}$ and $\mathcal{B}$ wins the EUF-CMA game.

Note that by the perfect ZK property of $\Pi^{\mathcal{R}_\mathsf{VK}}(\mathsf{opk})$ the simulation of the $\mathcal{O}^{\mathsf{Issue}}$ oracle is perfect. Moreover, the simulation of the $\mathcal{O}^{\mathtt{KU}+}$, the $\mathcal{O}^{\mathsf{ObtIss}}$ and the $\mathcal{O}^{\mathsf{Show}}$ oracles is perfect: In contrast to its definition, the simulated $\mathcal{O}^{\mathsf{ObtIss}}$ oracle does not create a credential, which goes unnoticed by $\mathcal{A}$ as the oracle returns only $\top$ or $\bot$ depending on whether the run was successful or not. Furthermore, the simulated $\mathcal{O}^{\mathtt{KU}+}$ and $\mathcal{O}^{\mathsf{Show}}$ oracles adapt their behaviors accordingly by creating missing credentials on the fly when needed.

Type 2. $\mathcal{B}$ interacts with the challenger $\mathcal{C}$ in the subset-soundness game of the scheme SC for some $t > 0$. We describe how $\mathcal{B}$ simulates the environment for $\mathcal{A}$ and interacts with $\mathcal{C}$. First, $\mathcal{C}$ generates set-commitment parameters $\mathsf{pp}_\mathsf{sc} \leftarrow (\mathsf{BG}, (a^i P, a^i \hat{P})_{i \in [t]})$ with $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P}) = \mathsf{BGGen}_\mathcal{R}(1^\kappa)$ and sends pp_sc to $\mathcal{B}$. $\mathcal{B}$ generates a key pair $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}_\mathcal{R}(\mathsf{BG}, 1^\ell)$ for $\ell = 3$, sets $(\mathsf{osk}, \mathsf{opk}) \leftarrow ((\bot, \mathsf{sk}), (\mathsf{pp}_\mathsf{sc}, \mathsf{pk}))$ and runs $\mathcal{A}(\mathsf{opk})$, simulating the oracles. All oracles are as in the real game, except for $\mathcal{O}^{\mathsf{Issue}}$, which is simulated as follows (note that $\mathcal{B}$ does not know a):

$\mathcal{O}^{\mathsf{Issue}}(i, \mathtt{A})$: The oracle is simulated as prescribed except for running the simulator $\mathcal{S}$ for $\Pi^{\mathcal{R}_\mathsf{VK}}(\mathsf{opk})$ (as it does not know $a = \mathsf{osk}[1]$).

By the perfect ZK property of $\Pi^{\mathcal{R}_\mathsf{VK}}(\mathsf{opk})$ the simulation of the $\mathcal{O}^{\mathsf{Issue}}$ oracle is perfect.

When $\mathcal{A}$ outputs $(\mathtt{A}'^*, \mathsf{st})$, $\mathcal{B}$ runs $\mathcal{A}(\mathsf{st})$ and interacts with $\mathcal{A}$ as verifier in a showing protocol. Assume $\mathcal{A}$ delivers a valid showing using $((C_1^*, C_2^*, C_3^*), \sigma^*)$ and a witness W^* for the attribute set $\mathtt{A}'^*$ such that $\mathtt{A}'^* \not\subseteq \mathtt{ATTR}[j]$ for all j with $\mathtt{I2U}[j] \in \mathtt{KU} \cup \mathtt{CU}$ and by conducting $\Pi^{\mathcal{R}_\mathsf{F}}(C_1^*, C_2^*, C_3^*)$. Then $\mathcal{B}$ runs the knowledge extractor of $\Pi^{\mathcal{R}_\mathsf{F}}$ to obtain a witness $w = (r'', \mu)$ such that $C_3^* = \mu P$. Let $(C', R', P) = \mu^{-1} \cdot (C_1^*, C_2^*, C_3^*)$; if there is no credential $\bot \neq ((C', R'), \sigma', *, *) \in \mathtt{CRED}$, then $\mathcal{B}$ aborts (the forgery was of Type 1). Otherwise, let j^* be such that $((C', R'), \sigma', r', O') = \mathtt{CRED}[j^*]$. If $\mathtt{I2U}[j^*] \in \mathtt{HU}$, then $\mathcal{B}$ aborts (the forgery was of Type 3). Else, we have $\mathtt{I2U}[j^*] \in \mathtt{KU} \cup \mathtt{CU}$ and $\mathtt{A}'^* \not\subseteq \mathtt{ATTR}[j^*]$. If for some $a' \in \mathtt{ATTR}[j^*]$ it holds that $a'P = aP$, then $\mathcal{B}$ sets $O^* \leftarrow (1, a', \mathtt{ATTR}[j^*])$. Else, $\mathcal{B}$ sets $O^* \leftarrow (0, \mu \cdot \mathtt{USK}[\mathtt{I2U}[j^*]], \mathtt{ATTR}[j^*])$. $\mathcal{B}$ outputs $(C_1^*, O^*, \mathtt{A}'^*, W^*)$, which satisfies $\mathtt{A}'^* \not\subseteq \mathtt{ATTR}[j^*] \neq \bot$ and $\mathsf{VerifySubset}(\mathsf{pp}_\mathsf{sc}, C_1^*, \mathtt{A}'^*, W^*) = 1$. $\mathcal{B}$'s output breaks thus subset soundness of SC.

Type 3. We assume the forgery to be of Type 3 and use a sequence of games which are indistinguishable under q-co-DL. Henceforth, we denote the event that an adversary wins Game i by S_i.

Game 0: The original game, which only outputs 1 if the forgery is of Type 3.

Game 1: As Game 0, except for the following oracles:

$\mathcal{O}^{\mathsf{ObtIss}}(i, \mathtt{A})$: As in Game 0, except that the experiment aborts if set-commitment trapdoor $a \in \mathtt{A}$.

$\mathcal{O}^{\mathsf{Issue}}(i, \mathtt{A})$: Analogous to the $\mathcal{O}^{\mathsf{ObtIss}}$ oracle.

Game 0 → Game 1: If $\mathcal{A}$ queries a set $\mathtt{A}$ with $a \in \mathtt{A}$ to one of the two oracles, then this breaks the q-co-DL assumption for $q = t$ and $\mathsf{BG} = \mathsf{BGGen}_{\mathcal{R}}(1^{\kappa})$. We have that $|\Pr[S_0] - \Pr[S_1]| \leq \epsilon_{qDL}(\kappa)$, where $\epsilon_{qDL}(\kappa)$ is the advantage of solving the q-co-DL assumption.

Game 2: As Game 1, with the difference that the $\mathcal{O}^{\mathsf{Show}}$ oracle is being run as follows:

$\mathcal{O}^{\mathsf{Show}}(j, \mathtt{A}')$: As in Game 0, but the ZKPoK $\Pi^{\mathcal{R}_{\mathsf{F}}}(C_1, C_2, C_3)$ is simulated via simulator $\mathcal{S}$.

Game 1 → Game 2: By the perfect ZK property of $\Pi^{\mathcal{R}_{\mathsf{F}}}$, we have that $\Pr[S_1] = \Pr[S_2]$.

Game 3: As Game 2, except that oracle $\mathcal{O}^{\mathtt{HU}+}$ is run as follows:

$\mathcal{O}^{\mathtt{HU}+}(i)$: As in Game 0, but when executing $\mathsf{UserKeyGen}(1^{\kappa})$, the experiment draws $\mathsf{usk} \xleftarrow{R} \mathbb{Z}_p$ instead of $\mathsf{usk} \xleftarrow{R} \mathbb{Z}_p^*$ and it aborts if $\mathsf{usk} = 0$.

Game 2 → Game 3: We have that $|\Pr[S_2] - \Pr[S_3]| \leq \frac{q_u}{p}$, where q_u is the number of queries to the $\mathcal{O}^{\mathtt{HU}+}$ oracle.

Game 4: As Game 3, except for the following changes. When $\mathcal{A}$ eventually delivers a valid showing by conducting the ZKPoK $\Pi^{\mathcal{R}_{\mathsf{F}}}(C_1^*, C_2^*, C_3^*)$, then the experiment runs the knowledge extractor $\mathcal{E}$ of $\Pi^{\mathcal{R}_{\mathsf{F}}}(C_1^*, C_2^*, C_3^*)$ and extracts a witness w.

Game 3 → Game 4: This change is only conceptual and we have $\Pr[S_3] = \Pr[S_4]$.

Game 5: As Game 4, except that we pick an index $k \xleftarrow{R} [q_o]$, where q_o is the number of queries to the $\mathcal{O}^{\mathsf{ObtIss}}$ oracle. The extracted witness w is such that $w = (r, \mu) \in (\mathbb{Z}_p^*)^2$ and $C_2^* = rC_1^*$ and $C_3^* = \mu P$ and if credential $((C', R'), \sigma', r', O') \leftarrow \mathtt{CRED}[k]$ is such that $(C', R', P) \neq \mu^{-1} \cdot (C_1^*, C_2^*, C_3^*)$, then the experiment aborts. Furthermore, we change the executions of the following oracle:

$\mathcal{O}^{\mathtt{KU}+}(i)$: As in Game 0, except that the experiment aborts when $i = \mathtt{I2U}[k]$.

Game 4 → Game 5: Note that when the forgery is of Type 3 then there exists some j s.t. for $\mathtt{CRED}[j] = ((C', R'), \sigma', r', O')$ we have $(C', R', P) = \mu^{-1} \cdot (C_1^*, C_2^*, C_3^*)$; moreover, $\mathtt{I2U}[j] \in \mathtt{HU}$. With probability $\frac{1}{q_o}$ we have $k = j$, in which case the experiment does not abort, i.e., we have $\Pr[S_5] \geq \frac{1}{q_o} \Pr[S_4]$.

We will now show that $\Pr[S_5] \leq \epsilon_{DL}(\kappa)$, where $\epsilon_{DL}(\kappa)$ is the advantage of solving the DLP. $\mathcal{B}$ plays the role of the challenger for $\mathcal{A}$ in Game 5 and obtains an instance (BG, xP) with $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P}) = \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$ for DLP in $\mathbb{G}_1$, generates $\mathsf{pp_{sc}} \leftarrow (\mathsf{BG}, (a^i P, a^i \hat{P})_{i\in[t]})$ by picking $a \xleftarrow{R} \mathbb{Z}_p$, generates $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^\ell)$ for $\ell = 3$ and sets $(\mathsf{osk}, \mathsf{opk}) \leftarrow ((a, \mathsf{sk}), (\mathsf{pp_{sc}}, \mathsf{pk}))$. Then, $\mathcal{B}$ runs $\mathcal{A}(\mathsf{opk})$ and simulates the oracles as in Game 5, except for the $\mathcal{O}^{\mathsf{Obtlss}}$ oracle, whose simulation deviates from Game 5 as follows:

$\mathcal{O}^{\mathsf{Obtlss}}(i, \mathtt{A})$: Let this be the jth query. $\mathcal{B}$ first computes $C \leftarrow \mathtt{USK}[i] \cdot f_{\mathtt{A}}(a) \cdot P$. If $j = k$ then it sets $R \leftarrow \mathtt{USK}[i] \cdot f_{\mathtt{A}}(a) \cdot xP$ $(= x \cdot C)$, $O = (0, \mathtt{USK}[i])$ and appends $\mathsf{cred} = ((C, R), \sigma, \bot, O)$ to $\mathtt{CRED}$. Otherwise $\mathcal{B}$ proceeds as in Game 5.

Note that since Game 2, the third component of the credential is not required to simulate $\mathcal{O}^{\mathsf{Show}}$ queries. When $\mathcal{A}$ outputs $(\mathtt{A}'^*, \mathsf{st})$, then $\mathcal{B}$ runs $\mathcal{A}(\mathsf{st})$ and interacts with $\mathcal{A}$ as verifier in a showing protocol. If $\mathcal{A}$ wins Game 5 using (C_1^*, C_2^*, C_3^*) and conducting the ZKPoK $\Pi^{\mathcal{R}_{\mathsf{F}}}(C_1^*, C_2^*, C_3^*)$, then $\mathcal{B}$ runs the knowledge extractor $\mathcal{E}$ of $\Pi^{\mathcal{R}_{\mathsf{F}}}(C_1^*, C_2^*, C_3^*)$ and extracts a witness $w = (r', \mu) \in (\mathbb{Z}_p^*)^2$ such that $C_2^* = r' C_1^*$ and $C_3^* = \mu P$. Further, we have that $((C', R'), \sigma', \bot, O') = \mathtt{CRED}[k]$. In the end, $\mathcal{B}$ outputs r' as a solution to the DLP in $\mathbb{G}_1$.

In total, we have $\Pr[S_5] \leq \epsilon_{DL}(\kappa)$, and with $\Pr[S_4] \leq q_o \cdot \Pr[S_5]$ as well as $\Pr[S_3] = \Pr[S_4]$ we obtain:

$$\Pr[S_3] = \Pr[S_4] \leq q_o \cdot \Pr[S_5] \leq q_o \cdot \epsilon_{DL}(\kappa).$$

With $|\Pr[S_2] - \Pr[S_3]| \leq \frac{q_u}{p}$ and $\Pr[S_1] = \Pr[S_2]$, we obtain:

$$\Pr[S_1] = \Pr[S_2] \leq \Pr[S_3] + \frac{q_u}{p} \leq q_o \cdot \epsilon_{DL}(\kappa) + \frac{q_u}{p}.$$

Using $|\Pr[S_0] - \Pr[S_1]| \leq \epsilon_{qDL}(\kappa)$, we finally have:

$$\Pr[S_0] \leq \Pr[S_1] + \epsilon_{qDL}(\kappa) \leq q_o \cdot \epsilon_{DL}(\kappa) + \frac{q_u}{p} + \epsilon_{qDL}(\kappa),$$

where $q = t$; q_o and q_u are the number of queries to the $\mathcal{O}^{\mathsf{Obtlss}}$ and the $\mathcal{O}^{\mathtt{HU}+}$ oracle, respectively. □

Proof of Theorem 7.7

The proof idea is to define a sequence of games in the last of which the answers of oracle $\mathcal{O}^{LoR}$ are independent of the bit b. In particular, such an answer contains (C_1, C_2, C_3), σ' and the proof $\Pi^{\mathcal{R}_{\mathsf{F}}}(C_1, C_2, C_3)$. We first replace the signature σ' by a fresh signature (Game 2) and simulate the proof $\Pi^{\mathcal{R}_{\mathsf{F}}}$ (Game 3). In Games 5 and 6 we replace C_1 and C_2 by random elements. Since $C_3 = \mu \cdot P$ for $\mu \xleftarrow{R} \mathbb{Z}_p^*$, in the final game the adversary receives a fresh signature σ' on a random tuple (C_1, C_2, C_3) and a simulated proof, resulting in a game that is independent of b.

Proof. We now prove the anonymity of Scheme 9 using a sequence of games. We assume that adversary $\mathcal{A}$ at some point calls $\mathcal{O}^{LoR}$ for some $(j_0, j_1, \mathtt{A}')$ with $\mathtt{I2U}[j_0], \mathtt{I2U}[j_1] \in \mathtt{HU}$. This is w.l.o.g., as otherwise the bit b is perfectly hidden from $\mathcal{A}$.

Henceforth, we denote the event that an adversary wins Game i by S_i.

Game 0: The original game as given in Definition 7.4.

Game 1: As Game 0, except for the $\mathcal{O}^{\mathsf{Obtain}}$ oracle. On the first successful completion of the ZKPoK $\Pi^{\mathcal{R}_{\mathsf{VK}}}(\mathsf{opk})$ (of which there must be at least one by the above assumption), the experiment runs its knowledge extractor $\mathcal{E}$, which extracts a witness (w_1, w_2).

Game 0 → Game 1: This change is only conceptual and we have $\Pr[S_0] = \Pr[S_1]$.

Game 2: As Game 1, except that the experiment sets $a \leftarrow w_1$ and $\mathsf{sk} \leftarrow w_2$ and runs the $\mathcal{O}^{LoR}$ oracle as follows:

$\mathcal{O}^{LoR}(j_0, j_1, \mathtt{A}')$: As in Game 0, except that all executions of $\mathsf{ChgRep}_{\mathcal{R}}((C, r \cdot C, P), \sigma, \mu, \mathsf{pk})$ for credential $(C, \sigma, r, O) \leftarrow \mathtt{CRED}[j_b]$ and $\mu \xleftarrow{R} \mathbb{Z}_p^*$ are replaced by $(\mu \cdot (C, r \cdot C, P), \mathsf{Sign}_{\mathcal{R}}(\mu \cdot (C, r \cdot C, P), \mathsf{sk}))$.

Game 1 → Game 2: By soundness of $\Pi^{\mathcal{R}_{\mathsf{VK}}}$, we have $\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk}) = 1$, and by SPS-EQ's perfect adaptation of signatures (Definition 3.7), $\mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \mu, \mathsf{pk})$ and $(\mu M, \mathsf{Sign}_{\mathcal{R}}(\mu M, \mathsf{sk}))$ are identically distributed for all $M \in (\mathbb{G}_1^*)^3$. We thus have $\Pr[S_1] = \Pr[S_2]$.

Game 3: As Game 2, except that the experiment runs the $\mathcal{O}^{LoR}$ oracle as follows:

$\mathcal{O}^{LoR}(j_0, j_1, \mathtt{A}')$: As in Game 2, but the ZKPoK $\Pi^{\mathcal{R}_{\mathsf{F}}}(C_1^*, C_2^*, C_3^*)$ is simulated using its simulator $\mathcal{S}$.

Game 2 → Game 3: By perfect ZK of $\Pi^{\mathcal{R}_{\mathsf{F}}}$, we have that $\Pr[S_2] = \Pr[S_3]$.

Game 4: As Game 3, except for the following changes. Let q_u be the number of queries made to the $\mathcal{O}^{\mathtt{HU}+}$ oracle. At the beginning Game 4 picks $k \xleftarrow{R} [q_u]$ and runs the $\mathcal{O}^{\mathtt{KU}+}$ and $\mathcal{O}^{LoR}$ oracles as follows.

$\mathcal{O}^{\mathtt{KU}+}(i)$: If $i \notin \mathtt{HU}$ or $i \in I_{LoR}$, it returns $\perp$ (as in the previous games). If $i = k$ then the experiment stops and outputs a random bit $b' \xleftarrow{R} \{0, 1\}$; otherwise it returns user i's usk and credentials and moves i from $\mathtt{HU}$ to $\mathtt{KU}$.

$\mathcal{O}^{LoR}(j_0, j_1, \mathtt{A}')$: As in Game 3, except that if $k \neq \mathtt{I2U}[j_b]$ then the experiment stops and outputs $b' \xleftarrow{R} \{0, 1\}$.

Game 3 → Game 4: By assumption, $\mathcal{O}^{LoR}$ is called at least once with some input $(j_0, j_1, \mathtt{A}')$ with $\mathtt{I2U}[j_0], \mathtt{I2U}[j_1] \in \mathtt{HU}$. If $k = \mathtt{I2U}[j_b]$ then $\mathcal{O}^{LoR}$ does not abort and neither does $\mathcal{O}^{\mathtt{KU}+}$ (it cannot have been called on $\mathtt{I2U}[j_b]$ before that call to $\mathcal{O}^{LoR}$ (otherwise $\mathtt{I2U}[j_b] \notin \mathtt{HU}$), if called afterwards, it returns $\perp$, since $k \in I_{LoR}$).

Since $k = \texttt{I2U}[j_b]$ with probability $\frac{1}{q_u}$, the probability that the experiment does not abort is at least $\frac{1}{q_u}$, i.e., $\Pr[S_4] \geq \frac{1}{2} + \frac{1}{q_u} \cdot (\Pr[S_3] - \frac{1}{2})$.

Game 5: As Game 4, except for the $\mathcal{O}^{LoR}$ oracle:

$\mathcal{O}^{LoR}(j_0, j_1, \mathtt{A}')$: As in Game 4, except that in addition to $\mu \xleftarrow{R} \mathbb{Z}_p^*$, it randomly picks $C_1 \xleftarrow{R} \mathbb{G}_1^*$ and performs the showing using $\mathsf{cred}' \xleftarrow{R} ((C_1, r \cdot C_1, \mu \cdot P), \mathsf{Sign}_{\mathcal{R}}((C_1, r \cdot C_1, \mu \cdot P), \mathsf{sk}))$, with $r \leftarrow \texttt{CRED}[j_b][3]$, and $W \leftarrow \bot$ (if $a \in \mathtt{A}'$) or $W \leftarrow f_{\mathtt{A}'}(a)^{-1} \cdot C_1$ (if $a \notin \mathtt{A}'$).

Note that the only difference is the choice of C_1; W is distributed as in Game 4, in particular, if $a \notin \mathtt{A}'$, it is the unique element satisfying $\mathsf{VerifySubset}(\mathsf{pp}, C, \mathtt{A}', W)$.

Game 4 → Game 5: Let $(\mathsf{BG}, xP, yP, zP)$ be a DDH instance with $\mathsf{BG} = \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$. After initializing the environment, the simulation initializes a list $L \leftarrow \emptyset$. The oracles are being simulated as in Game 4, except for the subsequent oracles, which are simulated as follows:

$\mathcal{O}^{\texttt{HU}+}(i)$: As in Game 4, but if $i = k$ it sets $\texttt{USK}[i] \leftarrow \bot$ and $\texttt{UPK}[i] \leftarrow xP$. (We have thus implicitly set $\mathsf{usk} \leftarrow x$.)

$\mathcal{O}^{\mathsf{Obtain}}(i, \mathtt{A})$: As in Game 4, except for the computation of the following values if $i = k$. Let this be the jth call to this oracle. If $a \notin \mathtt{A}$, it computes C as $C \leftarrow f_{\mathtt{A}}(a) \cdot xP$ and sets $L[j] \leftarrow \bot$. If $a \in \mathtt{A}$ it picks $\rho \xleftarrow{R} \mathbb{Z}_p^*$, computes C as $C \leftarrow \rho \cdot xP$ and sets $L[j] \leftarrow \rho$. (In both cases C is thus distributed as in the original game.)

$\mathcal{O}^{\mathsf{Show}}(j, \mathtt{A}')$: As in Game 4, with the difference that if $\texttt{I2U}[j] = k$ and $a \notin \mathtt{A}'$ it computes the witness $W \leftarrow \mu f_{\mathtt{A} \setminus \mathtt{A}'}(a) \cdot xP$. ($W$ is thus distributed as in the original game.)

$\mathcal{O}^{LoR}(j_0, j_1, \mathtt{A}')$: As in Game 4, with the difference that it picks $s, t \xleftarrow{R} \mathbb{Z}_p$ and computes $Y' \leftarrow t \cdot yP + sP = y'P$ with $y' \leftarrow ty + s$, and $Z' \leftarrow t \cdot zP + s \cdot xP = (t(z - xy) + xy')P$. (If $z \neq xy$ then Y' and Z' are independently random.) It performs the showing using the following values (implicitly setting $\mu \leftarrow y'$):

- If $a \notin \texttt{ATTR}[j_b]$: $C_1 \leftarrow f_{\mathtt{A}}(a) \cdot Z'$ and $W \leftarrow f_{\mathtt{A}'}(a)^{-1} \cdot C_1$;
- If $a \in \texttt{ATTR}[j_b] \setminus \mathtt{A}'$: $C_1 \leftarrow \rho \cdot Z'$ with $\rho \leftarrow L[j_b]$ and $W \leftarrow f_{\mathtt{A}'}(a)^{-1} \cdot C_1$;
- If $a \in \mathtt{A}'$: $C_1 \leftarrow \rho \cdot Z'$ with $\rho \leftarrow L[j_b]$ and $W \leftarrow \bot$;

$C_2 \leftarrow r \cdot C_1$, $C_3 \leftarrow Y'$ and $r \leftarrow \texttt{CRED}[j_b][3]$.

Apart from an error event happening with negligible probability, we have simulated Game 4 if the DDH instance was valid and Game 5 otherwise. If $xP = 0_{\mathbb{G}_1}$, or if during the simulation of the $\mathcal{O}^{LoR}$ oracle $Y' = 0_{\mathbb{G}_1}$ or $Z' = 0_{\mathbb{G}_1}$ then the distribution of values is not as in one of the two games. Otherwise, we have implicitly set $\mathsf{usk} \leftarrow x$ and $\mu \leftarrow y'$ (for a fresh value y' at every call of $\mathcal{O}^{LoR}$). In case of a DDH instance, we have (depending on the case) $C_1 \leftarrow \mathsf{usk}\mu f_{\mathtt{A}}(a) \cdot P$ (or $C_1 = \rho \cdot x\mu \cdot P = \mu \cdot C$); otherwise C_1 is independently random.

Hence, $|\Pr[S_4] - \Pr[S_5]| \leq \epsilon_{DDH}(\kappa) + \frac{1}{p} + q_l \cdot \frac{2}{p}$, where $\epsilon_{DDH}(\kappa)$ is the advantage of solving the DDH problem and q_l the number of queries to the $\mathcal{O}^{LoR}$ oracle.

Game 6: As Game 5, except for the $\mathcal{O}^{LoR}$ oracle:

$\mathcal{O}^{LoR}(j_0, j_1, \mathtt{A}')$: As in Game 5, except that in addition to μ and C_1 it also picks $C_2 \xleftarrow{R} \mathbb{G}_1^*$ and performs the showing using $\mathsf{cred}' \xleftarrow{R} ((C_1, C_2, \mu \cdot P), \mathsf{Sign}_{\mathcal{R}}((C_1, C_2, \mu \cdot P), \mathsf{sk}))$ and W as in Game 5.

Game 5 → Game 6: Let $(\mathsf{BG}, xP, yP, zP)$ be a DDH instance with $\mathsf{BG} = \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$. After initializing the environment, the simulation initializes a list $L \leftarrow \emptyset$. The oracles are being simulated as in Game 5, except for the subsequent oracles, which are simulated as follows:

$\mathcal{O}^{\mathsf{Obtain}}(i, \mathtt{A})$: As in Game 5, except for the computation of the following values if $i = k$. Let this be the jth call to this oracle. It first picks $u \xleftarrow{R} \mathbb{Z}_p$ and sets $X' \leftarrow xP + u \cdot P$ and $L[j] \leftarrow u$. If $a \notin \mathtt{A}$, it computes $C \leftarrow f_{\mathtt{A}}(a) \cdot \mathtt{USK}[i] \cdot P$ and $R \leftarrow f_{\mathtt{A}}(a) \cdot \mathtt{USK}[i] \cdot X'$. If $a \in \mathtt{A}$, it picks $\rho \xleftarrow{R} \mathbb{Z}_p^*$ and computes $C \leftarrow \rho \cdot P$ and $R \leftarrow \rho \cdot X'$. In both cases it sets $r \leftarrow \perp$ (r is implicitly set to $r \leftarrow x' := x + u$ and C and $R = r \cdot C$ are distributed as in the original game; unless $X' = 0_{\mathbb{G}_1}$). Note that, since the ZKPoK in $\mathcal{O}^{\mathsf{Show}}$ is simulated, r is not used anywhere in the game.

$\mathcal{O}^{LoR}(j_0, j_1, \mathtt{A}')$: As in Game 5, with the difference that it fetches $u \leftarrow L[j_b]$, picks $s, t \xleftarrow{R} \mathbb{Z}_p$ and computes $Y' \leftarrow t \cdot yP + s \cdot P = y'P$ with $y' \leftarrow ty + s$, and $Z' \leftarrow t \cdot zP + s \cdot xP + ut \cdot yP + us \cdot P = (t(z - xy) + x'y')P$. It picks $\mu \xleftarrow{R} \mathbb{Z}_p^*$ and performs the showing using $C_1 \leftarrow Y'$, $C_2 \leftarrow Z'$ and $C_3 \leftarrow \mu \cdot P$. Witness W is computed from C_1 as in the previous simulation.

Apart from an error event happening with negligible probability, we have simulated Game 5 if the DDH instance was valid and Game 6 otherwise. If $X' = 0_{\mathbb{G}_1}$ during the simulation of the $\mathcal{O}^{\mathsf{Obtain}}$ oracle, or if during the simulation of the $\mathcal{O}^{LoR}$ oracle $Y' = 0_{\mathbb{G}_1}$ or $Z' = 0_{\mathbb{G}_1}$ then the distribution of values is not as in one of the two games. Otherwise, we have implicitly set $r \leftarrow x'$ (for a fresh value x' at every call of $\mathcal{O}^{\mathsf{Obtain}}$) and $C_1 \leftarrow Y'$ (for a fresh value Y' at every call of $\mathcal{O}^{LoR}$). In case of a DDH instance, we have $C_2 = r \cdot C_1$ (as prescribed by Game 5); otherwise C_2 is independently random (as prescribed by Game 6).

Hence, $|\Pr[S_5] - \Pr[S_6]| \leq \epsilon_{DDH}(\kappa) + q_o \cdot \frac{1}{p} + q_l \cdot \frac{2}{p}$, where $\epsilon_{DDH}(\kappa)$ is the advantage of solving the DDH problem; q_o and q_l the number of queries to the $\mathcal{O}^{\mathsf{Obtain}}$ and $\mathcal{O}^{LoR}$ oracle, respectively.

In Game 6 the $\mathcal{O}^{LoR}$ oracle returns a fresh signature σ on a random triple $(C_1, C_2, C_3) \xleftarrow{R} (\mathbb{G}_1^*)^3$ and a simulated proof; bit b is thus information-theoretically hidden from $\mathcal{A}$. In total, we have

$$\Pr[S_5] \leq \Pr[S_6] + \epsilon_{DDH}(\kappa) + (q_o + 2q_l)\frac{1}{p} = \frac{1}{2} + \epsilon_{DDH}(\kappa) + (q_o + 2q_l)\frac{1}{p},$$

$$\Pr[S_4] \leq \Pr[S_5] + \epsilon_{DDH}(\kappa) + (1 + 2q_l)\frac{1}{p} \leq \frac{1}{2} + 2 \cdot \epsilon_{DDH}(\kappa) + (1 + q_o + 4q_l)\frac{1}{p}.$$

Then, we have that $\Pr[S_4] - \frac{1}{2} \geq \frac{1}{q_u} \cdot (\Pr[S_3] - \frac{1}{2})$ and $\Pr[S_0] = \Pr[S_1] = \Pr[S_2] = \Pr[S_3]$, giving us:

$$\Pr[S_0] = \Pr[S_3] \leq \frac{1}{2} + q_u \cdot \left(2 \cdot \epsilon_{DDH}(\kappa) + (1 + q_o + 4q_l)\frac{1}{p}\right).$$

where q_u, q_o and q_l are the number of queries to the $\mathcal{O}^{\mathsf{HU}+}$, $\mathcal{O}^{\mathsf{Obtain}}$ and the $\mathcal{O}^{LoR}$ oracle, respectively. □

7.2.6 A Concurrently Secure Scheme Variant

We now sketch a more efficient and concurrently secure variant of our scheme, which uses public parameters.

As briefly discussed in Section 2.6, [Dam00] proposes a generic transform which—under the assumption of OWFs and at the expense of a CRS—converts any Σ-protocol for an arbitrary NP-relation $\mathcal{R}$ into a 3-move concurrent ZKPoK (without any timing constraints). By introducing a setup algorithm and replacing the used ZKPoKs with those from [Dam00] (the statements proven stay the same), we obtain an ABC that is concurrently secure in the CRS model (and, in particular, anonymous under malicious organization keys in the CRS model) and uses only three moves during both issuing and showing (when interleaving the ZKPoK moves and the other protocol moves).

The introduction of system parameters pp further allows us to move the set-commitment parameters from the organization keys to pp, which reduces the organization public key sizes.

7.2.7 Efficiency Analysis and Comparison

We provide a brief comparison with other ABC approaches. As other candidates for multi-show ABCs, we take the Camenisch-Lysyanskaya schemes [CL01, CL03, CL04] as well as schemes from BBS$^+$ signatures [BBS04, ASM06] which cover a broad class of ABC schemes from randomizable signature schemes with efficient PoKs. Furthermore, we take two alternative multi-show ABC constructions [CL11, CL13] as well as Brands' approach [Bra00] (also covering the provable secure version [BL13a]) for the sake of completeness, although the latter only provides one-show ABCs. We omit other approaches such as [AMO08] that only allow a single attribute per credential. We also omit approaches that achieve more efficient showings for existing ABC systems only in very special cases such as for attribute values that come from a very small set (and are, thus, hard to compare).[4] Finally, we also include the recent approach in [CDHK15] that has the same asymptotic parameter sizes as our approach. They achieve security in

[4]For instance, the approach in [CG12] for CL credentials in the strong RSA setting (encoding attributes as prime numbers) or in a pairing-based setting using BBS$^+$ credentials [SNF11] (encoding attributes using accumulators) where the latter additionally requires very large public parameters (one F-secure BB signature [BCKL08] for every possible attribute value).

the UC framework [Can01], but consequently far less efficient constructions in a concrete setting. Their approach is equally expressive as ours (selective disclosure), but additionally supports pseudonyms and context-specific pseudonyms for showings. For our comparison in Table 7.1 we take their most efficient instantiation (which does not provide secret key extractability) and note that our showings require *less than 10 group elements* (when instantiated with Scheme 2 and the ZKPoK protocol from [CDM00]), whereas the cheapest variant in [CDHK15] requires *around 100 group elements*.

Table 7.1 gives an overview of these systems, where Type-1 and Type-2 refer to bilinear-group settings with Type-1 and Type-2 pairings, respectively. In a stronger sense, XDH as well as SXDH requires the respective assumption to hold. Furthermore, $\mathbb{G}_q$ denotes a group of prime order q (e.g., a subgroup of large order q of $\mathbb{Z}_p^*$ or an elliptic curve group of order q). By $|\mathbb{G}|$, we mean the bitlength of the representation of an element from group $\mathbb{G}$, by MK we indicate whether anonymity (privacy) holds with respect to maliciously generated issuer keys and with P we indicate whether the schemes support selective disclosure (s) or also proving relations about attributes (r). We note that $\circ$ indicates that the most efficient construction from [CDHK15] used in Table 7.1 does not consider malicious keys, while the other less efficient ones in [CDHK15] do.

We emphasize that, in contrast to other approaches, such as [CL04, CL13], our construction when instantiated with the SPS-EQ in Scheme 2 only requires a small and constant number of pairing evaluations in all protocol steps.

We stress that the model introduced in [CKL$^+$14] allows to instantiate constructions, for instance based on [CL03], that can deal with malicious organization keys (although at the cost of efficiency).

Table 7.1: Comparison of various approaches to ABC systems.

Scheme	Parameter Size (L attr.)			Issuing			Showing (k-of-L attr.)			MK	P
	Setting	CRS	Credential Size	Issuer	User	Comm	Verifier	User	Comm		
[CL03]	sRSA	$O(L)$	$O(1)$	$3\lvert\mathbb{Z}_N\rvert$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L-k)$	×	r
[CL04]	Type-1	$O(L)$	$O(L)$	$(2L+2)\lvert\mathbb{G}_1\rvert$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	×	r
[BBS04]	Type-2	$O(L)$	$O(1)$	$\lvert\mathbb{G}_1\rvert + 22\lvert\mathbb{Z}_q\rvert$	$O(L)$	$O(1)$	$O(L)$	$O(L)$	$O(L)$	×	r
[CL11]	Type-2	$O(1)$	$O(L)$	$L\lvert\mathbb{G}_1\rvert + 1\lvert\mathbb{G}_2\rvert$	$O(L)$	$O(L)$	$O(L)$	$O(1)$	$O(1)$	×	s
[CL13]	XDH	$O(L)$	$O(L)$	$(2L+2)(1\lvert\mathbb{G}_1\rvert + 1\lvert\mathbb{Z}_p\rvert)$	$O(L)$	$O(L)$	$O(k)$	$O(k)$	$O(k)$	×	s
[Bra00]	$\mathbb{G}_q$	$O(L)$	$O(1)$	$2\lvert\mathbb{G}_q\rvert + 2\lvert\mathbb{Z}_q\rvert$	$O(L)$	$O(1)$	$O(k)$	$O(k)$	$O(L-k)$	×	r
[CDHK15]	SXDH	$O(L)$	$O(1)$	$6\lvert\mathbb{G}_1\rvert + 2\lvert\mathbb{G}_2\rvert + 1\lvert\mathbb{Z}_p\rvert$	$O(L)$	$O(1)$	$O(k)$	$O(L-k)$	$O(1)$	○	s
Scheme 9	SXDH	$O(L)$	$O(1)$	$3\lvert\mathbb{G}_1\rvert + 1\lvert\mathbb{G}_2\rvert + 2\lvert\mathbb{Z}_p\rvert$	$O(L)$	$O(1)$	$O(k)$	$O(L-k)$	$O(1)$	✓	s

8 Conclusions

For everyone out there listening, thank you and Merry Christmas.
— Edward Snowden

In this thesis, we have introduced structure-preserving signatures on equivalence classes (SPS-EQs) and, based on this, we have demonstrated their potential by pointing out new ways to construct several practically efficient privacy-enhancing protocols and even a scheme for secure electronic-business processes. In particular, we have constructed practically efficient round-optimal blind signatures in the standard-model, an efficient standard-model one-show attribute-based credentials (ABCs), an efficient multi-show ABC protocol and a new verifiably-encrypted signature (VES) construction. Except for the VES scheme, all these schemes can be instantiated with any SPS-EQ that is perfectly adapting (under malicious keys)—a property on the distribution of signatures. In order to commit to the sets of attributes and present subsets during the showings, our multi-show ABC construction employs a set-commitment scheme. We have introduced the notion of set commitments along with a security model and presented a perfectly hiding set-commitment scheme. Note that these contributions are of independent interest.

With regard to SPS-EQ, we have introduced a security model and given two SPS-EQ constructions. Besides one scheme that fulfills all requirements for use in the presented applications while having only security guarantees in the generic-group model (GGM), we have also given a first standard-model scheme that is, however, not perfectly adapting. Nevertheless, it still fulfills a weaker property. Furthermore, we have seen an impossibility result, which provides evidence that the construction of a malicious-key perfectly adapting SPS-EQ in the standard-model is not trivial at all. This has motivated the definition of a

weaker unforgeability model, which is still sufficient for most applications and especially for the ones we have seen in this thesis.

Our blind-signature scheme is the first practically efficient round-optimal construction having proofs in the standard model. Round optimality is a distinguishing measure for efficiency and, moreover, guarantees concurrent security which, otherwise, has to be dealt with separately. Its main caveat is, however, that its blindness relies on an interactive (yet very plausible) assumption. Our scheme can be easily augmented to build partially blind signatures—the first of its kind in the standard model—and allows us to build efficient one-show ABCs with security guarantees in the standard model—the first such construction.

Our multi-show ABC, which is built from SPS-EQ and the introduced set commitment scheme, is the first ABC to achieve constant-size credentials and constant-time communication effort—both are important measures for (practical) efficiency. Another highlight is that it is the first scheme that is anonymous against malicious organization keys in the standard model. However, we achieve the former goals at the expense of reduced expressiveness, that is, we lose the ability to prove arbitrary relations about the attributes. Yet, we do not expect this to be a big drawback in many practical scenarios, as AND-relations can be proven and common relations (such as an age range) can still be encoded into the attributes. Furthermore, to prove our ABC secure, we have introduced a security model, which is the first comprehensive game-based security model and as a such also of independent interest.

Finally, we have also seen a new standard-model VES construction from SPS-EQ. It gives us important theoretical insights: As it is black-box, it allows us to relate SPS-EQs to public-key encryption and, in doing so, to separate certain types of SPS-EQ from one-way functions (OWFs). This relation is somewhat surprising, since digital signatures (DSs) can usually be built from OWFs. Last but not least, we have pointed out flaws in the VES security model and showed how to resolve them. To this end, we have given a secure VES having an underlying DS scheme that is neither correct nor unforgeable.

8.1 Open Issues and Future Work

An important issue left open is the construction of a (malicious-key) perfectly adapting SPS-EQ in the standard model. We recall that the standard-model construction in Scheme 3 requires a q-type assumption and only provides a weaker form of privacy. This would allow us to instantiate all discussed schemes and protocols in the standard model. Nevertheless, the impossibility result given in Section 3.6 gives us the necessary direction. Furthermore, it is an interesting question whether such signatures when built for other more general equivalence relations yield further interesting, alternative applications. Another open issue is to get rid of the interactive blindness assumption in our blind-signature scheme from Chapter 4. Future work regarding the application of SPS-EQs to anonymous credentials includes the investigation of their suitability to build delegatable anonymous credentials [CL06]. Eventually, there seem to be further

applications of SPS-EQs—especially to privacy-enhancing cryptography. Thus, for instance, it seems possible to build group signatures from SPS-EQ and it remains an open issue to develop further constructions of schemes and protocols.

Last but not least, an open issue is to develop software and hardware implementations of our schemes and to contrast them with existing implementations of alternative approaches. In particular, it would be interesting to compare implementations of our one-show and multi-show ABCs with Microsoft's U-Prove [Bra00] and IBM's Idemix [CV02].

A Omitted Proofs

A.1 Proof of Theorem 3.13

The following proof is taken in large parts verbatim from [FHS14].

We first consider the messages submitted to the signing oracle and the forgery output by the adversary as formal multivariate Laurent polynomials whose variables correspond to the secret values chosen by the challenger, and show that an adversary is unable to symbolically produce an existential forgery (even when message elements are adaptively chosen).

Then, in the second part we show that the probability for an adversary to produce an existential forgery by incident is negligible.

When proving the existentially unforgeable under adaptive chosen-message attacks (EUF-CMA) security of a structure-preserving signature (SPS) scheme, we have to take into account that an adversary is allowed to incorporate already queried signatures into new signature queries. Therefore, the degree of involved polynomials grows linearly in the number signature queries.

The values chosen by the challenger in the unforgeability game, which are unknown to the adversary, are $x_1, \ldots, x_\ell$ used in the public keys $(\hat{X}_i)_{i\in[\ell]} \in (\mathbb{G}_2^*)^\ell$ and the values y_j, $j \in [q]$, picked for the jth signature, that is, when the jth signing query for a message $(M_{j,i})_{i\in[\ell]}$ is answered as

$$(Z_j, Y_j, \hat{Y}_j) = (y_j \textstyle\sum_{i\in[\ell]} x_i M_{j,i}, \frac{1}{y_j} P, \frac{1}{y_j} \hat{P}).$$

When outputting a forgery $(Z^*, Y^*, \hat{Y}^*)$ for a message $(M_i^*)_{i\in[\ell]}$, the elements the adversary has seen are $(Z_j, Y_j)_{j\in[q]}$ in $\mathbb{G}_1$, and $(\hat{Y}_j)_{j\in[q]}$ as well as $(\hat{X}_i)_{i\in[\ell]}$

in $\mathbb{G}_2$. The forgery must thus have been computed by choosing

$$\pi_z, \pi_y, \pi_{\hat{y}}, \pi_{m^*,i}, \rho_{z,j}, \rho_{y,j}, \rho_{m^*,i,j}, \psi_{y,j}, \psi_{\hat{y},j}, \psi_{m^*,i,j}, \chi_{\hat{y},i} \in \mathbb{Z}_p \quad \text{for} \quad j \in [q], i \in [\ell]$$

and setting

$$\begin{aligned}
Z^* &= \pi_z P + \sum_{j\in[q]} \rho_{z,j} Z_j + \sum_{j\in[q]} \psi_{z,j} Y_j \\
Y^* &= \pi_y P + \sum_{j\in[q]} \rho_{y,j} Z_j + \sum_{j\in[q]} \psi_{y,j} Y_j \\
\hat{Y}^* &= \pi_{\hat{y}} \hat{P} + \sum_{i\in[\ell]} \chi_{\hat{y},i} \hat{X}_i + \sum_{j\in[q]} \psi_{\hat{y},j} \hat{Y}_j \\
M_i^* &= \pi_{m^*,i} P + \sum_{j\in[q]} \rho_{m^*,i,j} Z_j + \sum_{j\in[q]} \psi_{m^*,i,j} Y_j
\end{aligned}$$

Similarly, for all $j \in [q]$ the message $(M_{j,i})_{i\in[\ell]}$ submitted in the jth query is computed as a linear combination of all the $\mathbb{G}_1$ elements the adversary has seen so far, that is,

$$P, Z_1, Y_1, \ldots, Z_{j-1}, Y_{j-1}.$$

By considering all these group elements and taking their discrete logarithms to the bases P and $\hat{P}$, respectively, we obtain the following linear combinations:

$$\begin{aligned}
z^* &= \pi_z + \sum_{j\in[q]} \rho_{z,j} z_j + \sum_{j\in[q]} \psi_{z,j} \frac{1}{y_j} \\
y^* &= \pi_y + \sum_{j\in[q]} \rho_{y,j} z_j + \sum_{j\in[q]} \psi_{y,j} \frac{1}{y_j} \\
\hat{y}^* &= \pi_{\hat{y}} + \sum_{i\in[\ell]} \chi_{\hat{y},i} x_i + \sum_{j\in[q]} \psi_{\hat{y},j} \frac{1}{y_j} \\
m_i^* &= \pi_{m^*,i} + \sum_{j\in[q]} \rho_{m^*,i,j} z_j + \sum_{j\in[q]} \psi_{m^*,i,j} \frac{1}{y_j} \\
m_{j,i} &= \pi_{m,j,i} + \sum_{k\in[j-1]} \rho_{m,j,i,k} z_k + \sum_{k\in[j-1]} \psi_{m,j,i,k} \frac{1}{y_k}
\end{aligned}$$

Observe that all message elements as well as the elements $Y^*, \hat{Y}^*$ of the forgery must be different from $0_{\mathbb{G}_1}$ and $0_{\mathbb{G}_2}$, respectively, by definition. Plugging the forgery into the verification relations yields:

$$\prod_{i\in[\ell]} e(M_i^*, \hat{X}_i) = e(Z^*, \hat{Y}^*) \quad \wedge \quad e(Y^*, \hat{P}) = e(P, \hat{Y}^*)$$

and taking discrete logarithms to the basis $e(P, \hat{P})$ in $\mathbb{G}_T$, we obtain the following

equations:

$$\sum_{i\in[\ell]} m_i^* x_i = z^* \hat{y}^* \tag{A.1}$$

$$y^* = \hat{y}^* \tag{A.2}$$

The values m_i^*, z^*, $\hat{y}^*$, y^* are multivariate Laurent polynomials of total degree $O(q)$ in $x_1, \ldots, x_\ell, y_1, \ldots, y_q$. Our further analysis will be simplified by the following fact.

Claim A.1. *For all $n \geq 1$, the monomials that constitute z_n have the form*

$$\frac{1}{y_s^b} \prod_{k\in[t]} y_{j_k} \prod_{k\in[t]} x_{i_k} \tag{A.3}$$

with $1 \leq t \leq n$; for all $k_1 \neq k_2$: $j_{k_1} \neq j_{k_2}$; for all k: $j_k \leq n \wedge s < j_k$; $j_t = n$; and $b \in \{0,1\}$.

Proof. We prove the claim by induction.

$\underline{n = 1}$: As before the first signing query, the only element from $\mathbb{G}_1$ available to the adversary is P, we have $m_{1,i} = \pi_{m,1,i}$ and therefore

$$z_1 = \sum_{i\in[\ell]} \pi_{m,1,i} y_1 x_i,$$

which proves the base case.

$\underline{n \to n+1}$: Assume for all $k \in [n]$ the monomials of all z_k are of the form in (A.3). Since

$$m_{n+1,i} = \pi_{m,n+1,i} + \textstyle\sum_{k\in[n]} \rho_{m,n+1,i,k} z_k + \sum_{k\in[n]} \psi_{m,n+1,i,k} \frac{1}{y_k},$$

by the definition of $\mathsf{Sign}_{\mathcal{R}}$ we have

$$\begin{aligned} z_{n+1} = &\sum_{i\in[\ell]} \pi_{m,n+1,i}\, y_{n+1} x_i + \\ &\sum_{i\in[\ell]} \sum_{k\in[n]} \rho_{m,n+1,i,k}\, y_{n+1} z_k x_i + \\ &\sum_{i\in[\ell]} \sum_{k\in[n]} \psi_{m,n+1,i,k}\, y_{n+1} \frac{1}{y_k} x_i. \end{aligned}$$

The monomials in the first and the last sum are as claimed in the statement. By the induction hypothesis any monomial contained in any z_k is of the form $\frac{1}{y_s^b} \prod_{p\in[t]} y_{j_p} \prod_{p\in[t]} x_{i_p}$, with $t \leq n$, $j_t = k$ and $s < j_p$ for all j_p as well as $j_p < k$, for all j_p with $p < t$ (which are all different). Each such monomial leads thus to a monomial in the 2^{nd} sum in (A.1) of the

form $\frac{1}{y_s^b}\left(y_{n+1}\prod_{p\in[t]} y_{j_p}\right)\left(x_i\prod_{p\in[t]} x_{i_p}\right) = \frac{1}{y_s^b}\prod_{p\in[t']} y_{j_p}\prod_{p\in[t']} x_{i_p}$, with $t' \coloneqq t+1 \leq n+1$, $j_{t'} \coloneqq n+1$, $i_{t+1} \coloneqq i$. Moreover $t' \leq n+1$, all j_p are still different and $\leq n$ and $s < j_p$ for all j_p, which proves the induction step.

Together this proves the claim. □

We will in particular use that by Claim A.1 in any monomial in z_k there are always exactly as many y's as x's in the numerator and there are at least one y and one x; moreover there is at most one y in the denominator (and which does not cancel down). Moreover, we have:

Corollary A.2. *Any monomial can only occur in one unique z_n.*

Proof. This is implied by Claim A.1 as follows: For any monomial, let i^* be maximal such that the monomial contains y_{i^*}. Then the monomial does not occur in z_n with $n > i^*$, since z_n contains y_n contradicting maximality. It does not occur in z_n with $n < i^*$ either, since all y_j contained in z_n have $j \leq n$, meaning y_{i^*} does not occur in z_n; a contradiction. □

We start by investigating Equation (A.2):

$$y^* = \hat{y}^*$$
$$\pi_y + \sum_{j\in[q]} \rho_{y,j} z_j + \sum_{j\in[q]} \psi_{y,j}\frac{1}{y_j} = \pi_{\hat{y}} + \sum_{i\in[\ell]} \chi_{\hat{y},i} x_i + \sum_{j\in[q]} \psi_{\hat{y},j}\frac{1}{y_j}$$

By equating coefficients, and taking into account that by Claim A.1 no z_j contains monomials of the form $1, x_i$, or $\frac{1}{y_j}$, we obtain $\rho_{y,j} = 0$ for all $j \in [q]$ and

(i) $\pi_{\hat{y}} = \pi_y$

(ii) $\chi_{\hat{y},i} = 0 \quad \forall i \in [\ell]$

(iii) $\psi_{\hat{y},j} = \psi_{y,j} \quad \forall j \in [q]$

Let us now investigate Equation (A.1) (where in $\hat{y}^*$ we replace $\pi_{\hat{y}}, \chi_{\hat{y},i}$ and $\psi_{\hat{y},j}$ as per (i), (ii) and (iii), respectively):

$$\sum_{i\in[\ell]} m_i^* x_i \;=\; z^*\hat{y}^*$$

Filling in m_i^*, z^* and y^*, we obtain:

$$\sum_{i\in[\ell]} \Big(\pi_{m^*,i} + \sum_{j\in[q]} \rho_{m^*,i,j} z_j + \sum_{j\in[q]} \psi_{m^*,i,j}\frac{1}{y_j}\Big) x_i \;=$$
$$\Big(\pi_z + \sum_{j\in[q]} \rho_{z,j} z_j + \sum_{j\in[q]} \psi_{z,j}\frac{1}{y_j}\Big)\Big(\pi_y + \sum_{k\in[q]} \psi_{y,k}\frac{1}{y_k}\Big)$$

The RHS then expands to:

$$\pi_z\pi_y + \sum_{j\in[q]} \rho_{z,j}\pi_y\, z_j + \sum_{j\in[q]} \big(\psi_{z,j}\pi_y + \pi_z\psi_{y,j}\big)\frac{1}{y_j} + \sum_{(j,k)\in[q]^2} (\rho_{z,j}\psi_{y,k}\,\frac{1}{y_k} z_j + \psi_{z,j}\psi_{y,k}\,\frac{1}{y_j y_k}).$$

Equating coefficients for 1, we get:

(iv) $\pi_z\pi_y = 0$

Since by Claim A.1, no terms in $z_j x_i$, z_j and $\frac{1}{y_k} z_j$ are of the form $\frac{1}{y_j}$ or $\frac{1}{y_j y_k}$, equating coefficients for $\frac{1}{y_j}$ and $\frac{1}{y_j y_k}$ yields:

(v) $\psi_{z,j}\pi_y + \pi_z\psi_{y,j} = 0 \quad \forall j \in [q]$

(vi) $\psi_{z,j}\psi_{y,k} = 0 \quad \forall j,k \in [q]$

By (iv)–(vi), we have simplified Equation (A.1) to the following:

$$\sum_{i\in[\ell]} \Big(\pi_{m^*,i} + \sum_{j\in[q]} \rho_{m^*,i,j} z_j + \sum_{j\in[q]} \psi_{m^*,i,j}\frac{1}{y_j}\Big) x_i \;=\; \text{(A.4)}$$

$$\sum_{j\in[q]} \rho_{z,j}\pi_y\, z_j + \sum_{(j,k)\in[q]^2} \rho_{z,j}\psi_{y,k}\,\frac{1}{y_k} z_j. \qquad \text{(A.5)}$$

Let us analyze the monomials contained in the z_j's. By (A.3) in Claim A.1, there is an equal number of y's and x's in numerators of such monomials. Therefore, on the LHS the number of x's in all monomials is always greater than that of y's, meaning monomials of type (A.3) only occur on the RHS of (A.4).

We now show that $\rho_{z,n}\pi_y\, z_n = 0$ for all $n \in [q]$. Assume that for some $n \in [q]$ this is not the case. Since none of the monomials in z_n can appear on the LHS and by Corollary A.2, they do not appear in any other z_i, $i \neq n$, z_n must be subtracted by a term contained in $\frac{1}{y_k} z_j$ for some $j,k \in [q]$. The term in this z_j must not have y_k in the numerator, as otherwise it would cancel down and the number of y's and x's would be different, meaning it would not correspond to any monomial in z_n (which are of the form (A.3)). This also means that any monomial contained in z_n (in the first sum on the RHS) must have y_k in the denominator if it is to be equal to a term in $\frac{1}{y_k} z_j$.

Next, we observe that for $j = n$ monomials in z_n can only be equal to terms in $\frac{1}{y_k} z_j$. This is because the maximal i^* with y_{i^*} appearing in z_n would be different for any other z_j, $j \neq n$ (cf. the proof of Corollary A.2). But this means that any monomial in z_n, which by the above must have y_k in the denominator, also occurs in the z_n in the double sum, yielding a term with y_k^2 in the denominator. Since this cannot occur anywhere else in the equation by Corollary A.2, we arrived at a contradiction. We have thus:

(vii) $\rho_{z,j}\pi_y\, z_n = 0 \quad \forall j \in [q]$

Equation (A.1) has now the following, simplified representation:

$$\sum_{i\in[\ell]}\Big(\pi_{m^*,i}+\sum_{j\in[q]}\rho_{m^*,i,j}z_j+\sum_{j\in[q]}\psi_{m^*,i,j}\frac{1}{y_j}\Big)x_i=\sum_{(j,k)\in[q]^2}\rho_{z,j}\psi_{y,k}\,\frac{1}{y_k}z_j \quad \text{(A.6)}$$

From Claim A.1 we have that every monomial of z_j has an equal number of y's and x's in the numerator; for all monomials of the LHS we thus have: (number of y's) = (number of x's) $-$ 1. For such a term to occur on the RHS, this has to be a monomial N in z_j that has y_k in the numerator, so it cancels down and leads to a term with more x's than y's. We show that this must be z_k, that is, we show that $\rho_{z,j}\psi_{y,k}=0$ for all $j\neq k$.

First this holds for $k>j$, since the "largest" y contained in z_j is y_j and thus y_k does not cancel. Second for $k<j$, let us assume that there is at least one pair of coefficients $\rho_{z,j}\psi_{y,k}\neq 0$ with $k<j$. Observe that $\frac{1}{y_k}z_j$ on the RHS still contains y_j as "largest" y-value (by Claim A.1). The monomials composing $\frac{1}{y_k}z_j$ do thus only occur in z_j on the LHS, thus $\rho_{m^*,i,j}\neq 0$ for some $i\in[\ell]$. Thus the monomial N from z_j on the RHS which contains y_k also occurs on the LHS. However, as by Claim A.1 every y occurs only once in every monomial, after canceling out y_k from z_j no y_k remains in N on the RHS. As however, y_k is present in the corresponding monomial in z_j on the LHS, there is no corresponding term on the RHS. A contradiction. We thus obtain:

(viii) $\rho_{z,j}\psi_{y,k}=0 \quad \forall j,k\in[q], j\neq k$

Since the RHS of (A.6) cannot be 0 (otherwise all m_i^* on the LHS would be 0, which is not a valid forgery), we have:

(ix) $\exists\, k\in[q]:\rho_{z,k}\psi_{y,k}\neq 0$

We now argue that there exists exactly one such k, which follows from the following basic fact:

Claim A.3. *Let $a,b\in\mathbb{Z}_p^q$ be two non-zero vectors. If $C=a\cdot b^\top$ is a diagonal matrix then at most one element in C is non-zero.*

Proof. Since C is diagonal, we have

$$\mathrm{rank}(C)=\#(\text{non-zero rows in C})=\#(\text{non-zero elements in C}).$$

From basic linear algebra we have $\mathrm{rank}(a)=\mathrm{rank}(b^\top)=1$ and $\mathrm{rank}(C)\leq \min\{\mathrm{rank}(a),\mathrm{rank}(b^\top)\}=1$. □

Applying this to $C:=(\rho_{z,j})_{j\in[q]}\cdot(\psi_{y,k})_{k\in[q]}^\top$, which by (viii) and (ix) is a non-zero diagonal matrix, we get that all but one element of the diagonal $(\rho_{z,k}\psi_{y,k})_{k\in[q]}$ are zero, that is:

(x) $\exists!\, n\in[q]:\rho_{z,n}\psi_{y,n}\neq 0$

By (viii) and (x), Equation (A.1) simplifies to

$$\sum_{i\in[\ell]}\Big(\pi_{m^*,i}+\sum_{j\in[q]}\rho_{m^*,i,j}z_j+\sum_{j\in[q]}\psi_{m^*,i,j}\frac{1}{y_j}\Big)x_i =$$
$$\rho_{z,n}\psi_{y,n}\,\frac{1}{y_n}z_n =$$
$$\rho_{z,n}\psi_{y,n}\sum_{i\in[\ell]}m_{n,i}x_i =$$
$$\rho_{z,n}\psi_{y,n}\sum_{i\in[\ell]}\Big(\pi_{m,n,i}+\sum_{j\in[n-1]}\rho_{m,n,i,j}z_j+\sum_{j\in[n-1]}\psi_{m,n,i,j}\frac{1}{y_j}\Big)x_i,$$

where in the 2nd line we substituted z_n by its definition, i.e., $y_n\sum_{k\in[\ell]}m_{n,k}x_k$, and in the 3rd line we replaced $m_{n,i}$ by its definition. Since by Claim A.1, x_i, z_jx_i and $\frac{1}{y_j}x_i$, for all $i\in[\ell], j\in[q]$, do not have common monomials, equating coefficients yields (with $\alpha \coloneqq \rho_{z,n}\psi_{y,n}$):

$$\pi_{m^*,i}=\alpha\,\pi_{m,n,i} \qquad \rho_{m^*,i,j}=\alpha\,\rho_{m,n,i,j} \qquad \psi_{m^*,i,j}=\alpha\,\psi_{m,n,i,j}$$

This finally means that the message for the forgery is just a multiple of the previously queried message M_n, which completes the first part of the proof.

It remains to be shown that the probability for an adversary to produce an existential forgery by "incident", i.e., that two formally different polynomials collide by evaluating to the same value (or, equivalently, that the difference polynomial evaluates to zero), is negligible. Suppose that the adversary makes q queries to the signing oracle and $O(q)$ queries to the group oracles. Then, all involved formal polynomials resulting from querying the group oracles are of degree $O(q)$ and overall there are $O(\binom{q}{2}) = O(q^2)$ polynomials that could collide (i.e., whose difference polynomial evaluates to zero). Then, by the Schwartz-Zippel lemma [Zip79, Sch80] and the collision argument, the probability of such an error in the simulation of the generic group is $O(q^3/p)$ and is, therefore negligible in the security parameter. □

A.2 Proof of Proposition 4.12

This proof is taken from [FHS15a].

Let $\mathcal{A}$ be a generic PPT adversary and let $\sigma\colon \mathbb{G}_1 \to \{0,1\}^{m_1}$, $\hat{\sigma}\colon \mathbb{G}_2 \to \{0,1\}^{m_2}$ and $\tau\colon \mathbb{G}_T \to \{0,1\}^{m_T}$ be random, homomorphic encoding functions where w.l.o.g. $m_1 < m_2 < m_T$. $\mathcal{A}$ cannot work directly with group elements, but is forced to work with their image under $\sigma, \hat{\sigma}$ and τ. Furthermore, $\mathcal{A}$ is given oracle access to perform generic bilinear-group operations (operations in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ and pairings). Since $\mathcal{A}$ is given access to the group-element encodings, it can perform equality checks on its own through string equality tests. At last, we require that $\mathcal{A}$ can only submit already queried encodings to the group oracles. (Note that we can enforce this by choosing m_1, m_2 and m_T large enough making

the probability of guessing bitstrings in the image of $\sigma, \hat{\sigma}$ and τ, respectively, negligible.)

Now, let $\mathcal{B}$ be an algorithm interacting with $\mathcal{A}$ as follows. $\mathcal{B}$ picks a random bit $b \xleftarrow{R} \{0,1\}$, picks $\sigma_P \xleftarrow{R} \{0,1\}^{m_1}$ and $\hat{\sigma}_{\hat{P}} \xleftarrow{R} \{0,1\}^{m_2}$ as encoding of the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. $\mathcal{B}$ stores $(1, \sigma_P)$ in a list L_1 and $(1, \hat{\sigma}_{\hat{P}})$ in a list L_2 and gives the respective encodings to $\mathcal{A}$. Furthermore, it initializes a list L_T to manage elements of $\mathbb{G}_T$. At first, $\mathcal{B}$ simulates the group oracles as follows.

Group action in $\mathbb{G}_1$: Given two bitstrings σ_0, σ_1 representing elements in $\mathbb{G}_1$, $\mathcal{B}$ looks them up in L_1 and recovers the first components $f_0, f_1 \in \mathbb{Z}_p$ of the corresponding entries (f_i, σ_i). It computes $f_0 + f_1$ and if L_1 already contains an entry starting with $f_0 + f_1$, $\mathcal{B}$ returns its associated bitstring σ; otherwise, $\mathcal{B}$ chooses $\sigma \xleftarrow{R} \{0,1\}^{m_1}$, returns σ and stores $(f_0 + f_1, \sigma)$ in L_1.

Inversion in $\mathbb{G}_1$: Given a bitstring σ representing an element in $\mathbb{G}_1$, $\mathcal{B}$ recovers the corresponding values $f \in \mathbb{Z}_p$ and computes $-f$. In case L_1 already contains $-f$, $\mathcal{B}$ returns its associated bitstring σ'. Otherwise, $\mathcal{B}$ chooses $\sigma' \xleftarrow{R} \{0,1\}^{m_1}$, returns σ' and stores $(-f, \sigma')$ in L_1.

Group action in $\mathbb{G}_2$: Given two bitstrings $\hat{\sigma}_0, \hat{\sigma}_1$ representing elements in $\mathbb{G}_2$, $\mathcal{B}$ recovers the corresponding values $\hat{f}_0, \hat{f}_1 \in \mathbb{Z}_p$ and computes $\hat{f}_0 + \hat{f}_1$. In case L_2 already contains $\hat{f}_0 + \hat{f}_1$, $\mathcal{B}$ returns its associated bitstring $\hat{\sigma}$. Otherwise, $\mathcal{B}$ chooses $\hat{\sigma} \xleftarrow{R} \{0,1\}^{m_2}$, returns $\hat{\sigma}$ and stores $(\hat{f}_0 + \hat{f}_1, \hat{\sigma})$ in L_2.

Inversion in $\mathbb{G}_2$: Given a bitstring $\hat{\sigma}$ representing an element in $\mathbb{G}_2$, $\mathcal{B}$ recovers the corresponding values $\hat{f} \in \mathbb{Z}_p$ and computes $-\hat{f}$. In case L_2 already contains $-\hat{f}$, $\mathcal{B}$ returns its associated bitstring $\hat{\sigma}'$. Otherwise, $\mathcal{B}$ chooses $\hat{\sigma}' \xleftarrow{R} \{0,1\}^{m_2}$, returns $\hat{\sigma}'$ and stores $(-\hat{f}, \hat{\sigma}')$ in L_2.

Pairing: Given two bitstrings $\sigma, \hat{\sigma}$ representing elements in $\mathbb{G}_1$ and $\mathbb{G}_2$, $\mathcal{B}$ recovers the corresponding values f from L_1 and $\hat{f}$ from L_2. In case L_T already contains $f \cdot \hat{f}$, $\mathcal{B}$ returns its associated bitstring τ. Otherwise, $\mathcal{B}$ chooses $\tau \xleftarrow{R} \{0,1\}^{m_T}$, returns τ and stores $(f \cdot \hat{f}, \tau)$ in L_T.

The group action and inversion oracle for $\mathbb{G}_T$ are simulated analogously to those for $\mathbb{G}_1$ and $\mathbb{G}_2$.

When $\mathcal{A}$ publishes σ_Q and $\hat{\sigma}_{\hat{Q}}$ such that $(f_Q, \sigma_Q) \in L_1$ and $(\hat{f}_{\hat{Q}}, \hat{\sigma}_{\hat{Q}}) \in L_2$ and $f_Q = \hat{f}_{\hat{Q}}$, $\mathcal{B}$ chooses four bitstrings $\sigma_0, \sigma_1, \sigma_2, \sigma_3 \xleftarrow{R} \{0,1\}^{m_1}$ and assigns polynomials $R, f_Q \cdot R, S, f_Q \cdot ((1-b) \cdot T + b \cdot U) \in \mathbb{Z}_p[R, S, T, U]$ to these values (in that order) in order to keep track of them. $\mathcal{B}$ stores $(R, \sigma_0), (f_Q \cdot R, \sigma_1), (S, \sigma_2), (f_Q \cdot ((1-b) \cdot T + b \cdot U), \sigma_3)$ in L_1 and provides $\mathcal{A}$ with $\sigma_0, \sigma_1, \sigma_2, \sigma_3$.

After this, $\mathcal{B}$ simulates the $\mathbb{G}_1$ group oracles as follows.

Group action in $\mathbb{G}_1$: Given two bitstrings σ_0, σ_1 representing elements in $\mathbb{G}_1$, $\mathcal{B}$ recovers the corresponding polynomials $f_0, f_1 \in \mathbb{Z}_p[R, S, T, U]$ and computes $f_0 + f_1$. In case L_1 already contains $f_0 + f_1$, $\mathcal{B}$ returns its associated bitstring. Otherwise, $\mathcal{B}$ chooses $\sigma \xleftarrow{R} \{0,1\}^{m_1}$, returns σ and stores $(f_0 + f_1, \sigma)$ in L_1.

Inversion in $\mathbb{G}_1$: Given a bitstring σ representing an element in $\mathbb{G}_1$, $\mathcal{B}$ recovers the corresponding values $f \in \mathbb{Z}_p[R, S, T, U]$ and computes $-f$. In case L_1 already contains $-f$, $\mathcal{B}$ returns its associated bitstring. Otherwise, $\mathcal{B}$ chooses $\sigma' \xleftarrow{R} \{0,1\}^{m_1}$, returns σ' and stores $(-f, \sigma')$ in L_1.

The group oracles for $\mathbb{G}_T$ are modified analogously to handle polynomials in $\mathbb{Z}_p[R, S, T, U]$.

When $\mathcal{A}$ has finished querying the group oracles, $\mathcal{A}$ outputs a bit b^*. Then, $\mathcal{B}$ chooses $r, s, t \xleftarrow{R} \mathbb{Z}_p$ and sets $R \leftarrow r, S \leftarrow s, T \leftarrow t, U \leftarrow rs$.

Now, if the simulation was consistent, no information about b got revealed and hence $\mathcal{A}$ can only guess b with probability $1/2$. Nevertheless, the simulation can be inconsistent, if two distinct polynomials in L_1 or in L_T evaluate to the same value after choosing concrete values for R, S, T, U. Note that such collisions cannot occur in L_2, since L_2 contains only polynomials of degree 0.

We need to prove that such a collision in L_1 (and likewise in L_T) cannot be caused by $\mathcal{A}$ itself. All substitutions in the formal variables R, S, T are independent, whereas only U depends on R and S. Therefore, $\mathcal{A}$ can only produce collisions using RS. In the beginning, the list L_1 only contains polynomials of degree 0, whereas later polynomials of total degree 1 are being added to L_1. Moreover, the group oracles do not increase the degree of the polynomials in L_1 as they only cover addition and inversion. Thus, $\mathcal{A}$ is not able to generate such collisions on purpose.

The same argumentation holds for L_T. Observe that the polynomials contained in L_T have at most total degree 1, since they arise from the multiplication of degree-0 polynomials in L_2 and polynomials of total degree at most 1 in L_1.

What remains to be shown is that the probability of a collision is negligible, i.e., that two distinct polynomials in L_1 and L_T evaluate to the same value after the substitution (or alternatively that their difference polynomial evaluates to 0). Suppose that $\mathcal{A}$ has issued q queries to the group oracles. Let $|L_1| = O(q)$ and $|L_T| = O(q)$, then there are $O(\binom{q}{2})$ possibilities of colliding polynomials. By the Schwartz-Zippel lemma [Zip79, Sch80] and the collision argument, the probability of such an error in the simulation of the generic group is $O(q^2/p)$ and is therefore negligible in the security parameter. The same kind of argument also holds for $\mathbb{G}_T$. □

Bibliography

[ABC+12] Jae Hyun Ahn, Dan Boneh, Jan Camenisch, Susan Hohenberger, abhi shelat, and Brent Waters. Computing on authenticated data. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 1–20, Taormina, Sicily, Italy, March 19–21, 2012. Springer, Heidelberg, Germany. 9

[Abe01] Masayuki Abe. A secure three-move blind signature scheme for polynomially many signatures. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 136–151, Innsbruck, Austria, May 6–10, 2001. Springer, Heidelberg, Germany. 10, 11, 74, 103

[AFG+10] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 209–236, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Heidelberg, Germany. 5, 8, 10, 39, 40, 55

[AFH+15] Martin R. Albrecht, Pooya Farshim, Dennis Hofheinz, Enrique Larraia, and Kenneth G. Paterson. Multilinear maps from obfuscation. Cryptology ePrint Archive, Report 2015/780, 2015. `http://eprint.iacr.org/2015/780`. 5

[AGHO11] Masayuki Abe, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Optimal structure-preserving signatures in asymmetric bilinear groups. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 649–666, Santa Barbara, CA, USA, August 14–18, 2011. Springer, Heidelberg, Germany. 8, 40, 47, 49, 55, 57

[AGO11] Masayuki Abe, Jens Groth, and Miyako Ohkubo. Separating short structure-preserving signatures from non-interactive assumptions. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 628–646, Seoul, South Korea, December 4–8, 2011. Springer, Heidelberg, Germany. 8, 40, 47

[AGOT14a] 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, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 390–407, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany. 8, 40, 47

[AGOT14b] Masayuki Abe, Jens Groth, Miyako Ohkubo, and Mehdi Tibouchi. Unified, minimal and selectively randomizable structure-preserving signatures. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 688–712, San Diego, CA, USA, February 24–26, 2014. Springer, Heidelberg, Germany. 8, 40

[AHO10] Masayuki Abe, Kristiyan Haralambiev, and Miyako Ohkubo. Signing on elements in bilinear groups for modular protocol design. Cryptology ePrint Archive, Report 2010/133, 2010. `http://eprint.iacr.org/2010/133`. 8, 40

[AKOT15] Masayuki Abe, Markulf Kohlweiss, Miyako Ohkubo, and Mehdi Tibouchi. Fully structure-preserving signatures and shrinking commitments. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 35–65, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany. 8, 39

[ALP12] Nuttapong Attrapadung, Benoît Libert, and Thomas Peters. Computing on authenticated data: New privacy definitions and constructions. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 367–385, Beijing, China, December 2–6, 2012. Springer, Heidelberg, Germany. 9

[ALP13] Nuttapong Attrapadung, Benoît Libert, and Thomas Peters. Efficient completely context-hiding quotable and linearly homomorphic signatures. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013: 16th International Conference on Theory and Practice of Public Key Cryptography*, volume 7778 of *Lecture Notes in Computer Science*, pages 386–404, Nara, Japan, February 26 – March 1, 2013. Springer, Heidelberg, Germany. 9

[AMO08] Norio Akagi, Yoshifumi Manabe, and Tatsuaki Okamoto. An efficient anonymous credential system. In Gene Tsudik, editor, *FC 2008: 12th International Conference on Financial Cryptography and Data Security*, volume 5143 of *Lecture Notes in Computer Science*, pages 272–286, Cozumel, Mexico, January 28–31, 2008. Springer, Heidelberg, Germany. 121

[ANN06] Michel Abdalla, Chanathip Namprempre, and Gregory Neven. On the (im)possibility of blind message authentication codes. In David Pointcheval, editor, *Topics in Cryptology – CT-RSA 2006*, volume 3860 of *Lecture Notes in Computer Science*, pages 262–279, San Jose, CA, USA, February 13–17, 2006. Springer, Heidelberg, Germany. 63

[AO00] Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signatures. In Mihir Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 271–286, Santa Barbara, CA, USA, August 20–24, 2000. Springer, Heidelberg, Germany. 63, 64

[ASM06] Man Ho Au, Willy Susilo, and Yi Mu. Constant-size dynamic k-TAA. In Roberto De Prisco and Moti Yung, editors, *SCN 06: 5th International Conference on Security in Communication Networks*, volume 4116 of *Lecture Notes in Computer Science*, pages 111–125, Maiori, Italy, September 6–8, 2006. Springer, Heidelberg, Germany. 121

[ASW98] N. Asokan, Victor Shoup, and Michael Waidner. Optimistic fair exchange of digital signatures (extended abstract). In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 591–606, Espoo, Finland, May 31 – June 4, 1998. Springer, Heidelberg, Germany. 79

[BB04] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany. 26, 27, 28

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55, Santa Barbara, CA, USA, August 15–19, 2004. Springer, Heidelberg, Germany. 5, 8, 12, 26, 121, 123

[BC14] Dan Boneh and Henry Corrigan-Gibbs. Bivariate polynomials modulo composites and their applications. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 42–62, Kaoshiung, Taiwan, R.O.C., December 7–11, 2014. Springer, Heidelberg, Germany. 12

[BCC+09] Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. Randomizable proofs and

delegatable anonymous credentials. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 108–125, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Heidelberg, Germany. 6, 10, 12

[BCG+14] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474, Berkeley, California, USA, May 18–21, 2014. IEEE Computer Society Press. 2

[BCKL08] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. P-signatures and noninteractive anonymous credentials. In Ran Canetti, editor, *TCC 2008: 5th Theory of Cryptography Conference*, volume 4948 of *Lecture Notes in Computer Science*, pages 356–374, San Francisco, CA, USA, March 19–21, 2008. Springer, Heidelberg, Germany. 12, 121

[BDZ03] Feng Bao, Robert H. Deng, and Huafei Zhu. Variations of Diffie-Hellman problem. In Sihan Qing, Dieter Gollmann, and Jianying Zhou, editors, *ICICS 03: 5th International Conference on Information and Communication Security*, volume 2836 of *Lecture Notes in Computer Science*, pages 301–312, Huhehaote, China, October 10–13, 2003. Springer, Heidelberg, Germany. 26, 89

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Heidelberg, Germany. 4

[BFKW09] Dan Boneh, David Freeman, Jonathan Katz, and Brent Waters. Signing a linear subspace: Signature schemes for network coding. In Stanislaw Jarecki and Gene Tsudik, editors, *PKC 2009: 12th International Conference on Theory and Practice of Public Key Cryptography*, volume 5443 of *Lecture Notes in Computer Science*, pages 68–87, Irvine, CA, USA, March 18–20, 2009. Springer, Heidelberg, Germany. 9

[BFPV11] Olivier Blazy, Georg Fuchsbauer, David Pointcheval, and Damien Vergnaud. Signatures on randomizable ciphertexts. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011: 14th International Conference on Theory and Practice of Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 403–422, Taormina, Italy, March 6–9, 2011. Springer, Heidelberg, Germany. 8, 10

[BG93] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In Ernest F. Brickell, editor, *Advances in Cryptology –*

CRYPTO'92, volume 740 of *Lecture Notes in Computer Science*, pages 390–420, Santa Barbara, CA, USA, August 16–20, 1993. Springer, Heidelberg, Germany. 32

[BGLS03] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432, Warsaw, Poland, May 4–8, 2003. Springer, Heidelberg, Germany. 4, 6, 13, 17, 79, 80, 83

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In Joe Kilian, editor, *TCC 2005: 2nd Theory of Cryptography Conference*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341, Cambridge, MA, USA, February 10–12, 2005. Springer, Heidelberg, Germany. 25

[BKLS02] Paulo S. L. M. Barreto, Hae Yong Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 354–368, Santa Barbara, CA, USA, August 18–22, 2002. Springer, Heidelberg, Germany. 4

[BL13a] Foteini Baldimtsi and Anna Lysyanskaya. Anonymous credentials light. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 13: 20th Conference on Computer and Communications Security*, pages 1087–1098, Berlin, Germany, November 4–8, 2013. ACM Press. 11, 15, 101, 102, 103, 104, 121

[BL13b] Foteini Baldimtsi and Anna Lysyanskaya. On the security of one-witness blind signature schemes. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part II*, volume 8270 of *Lecture Notes in Computer Science*, pages 82–99, Bengalore, India, December 1–5, 2013. Springer, Heidelberg, Germany. 11, 102

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532, Gold Coast, Australia, December 9–13, 2001. Springer, Heidelberg, Germany. 4

[Blu81] Manuel Blum. Coin flipping by telephone. In Allen Gersho, editor, *Advances in Cryptology – CRYPTO'81*, volume ECE Report 82-04, pages 11–15, Santa Barbara, CA, USA, 1981. U.C. Santa Barbara, Dept. of Elec. and Computer Eng. 29

[BMV08] Emmanuel Bresson, Jean Monnerat, and Damien Vergnaud. Separation results on the "one-more" computational problems. In Tal Malkin, editor, *Topics in Cryptology – CT-RSA 2008*, volume 4964 of *Lecture Notes in Computer Science*, pages 71–87, San Francisco, CA, USA, April 7–11, 2008. Springer, Heidelberg, Germany. 23

[BN06] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In Bart Preneel and Stafford Tavares, editors, *SAC 2005: 12th Annual International Workshop on Selected Areas in Cryptography*, volume 3897 of *Lecture Notes in Computer Science*, pages 319–331, Kingston, Ontario, Canada, August 11–12, 2006. Springer, Heidelberg, Germany. 4, 5, 25, 66, 86, 111

[BNPS03] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215, June 2003. 10

[Bol03] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Yvo Desmedt, editor, *PKC 2003: 6th International Workshop on Theory and Practice in Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 31–46, Miami, USA, January 6–8, 2003. Springer, Heidelberg, Germany. 5, 10

[Boy08] Xavier Boyen. The uber-assumption family (invited talk). In Steven D. Galbraith and Kenneth G. Paterson, editors, *PAIRING 2008: 2nd International Conference on Pairing-based Cryptography*, volume 5209 of *Lecture Notes in Computer Science*, pages 39–56, Egham, UK, September 1–3, 2008. Springer, Heidelberg, Germany. 27, 28

[BP10] Stefan Brands and Christian Paquin. U-Prove Cryptographic Specification v1. 2010. 2, 6, 11, 102

[BPV12] Olivier Blazy, David Pointcheval, and Damien Vergnaud. Compact round-optimal partially-blind signatures. In Ivan Visconti and Roberto De Prisco, editors, *SCN 12: 8th International Conference on Security in Communication Networks*, volume 7485 of *Lecture Notes in Computer Science*, pages 95–112, Amalfi, Italy, September 5–7, 2012. Springer, Heidelberg, Germany. 10

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press. 23

[Bra00] Stefan Brands. *Rethinking Public-Key Infrastructures and Digital Certificates: Building in Privacy.* MIT Press, 2000. 2, 11, 101, 102, 109, 121, 123, 127

[BS02a] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. Cryptology ePrint Archive, Report 2002/080, 2002. `http://eprint.iacr.org/2002/080`. 5

[BS02b] Emmanuel Bresson and Jacques Stern. Proofs of knowledge for non-monotone discrete-log formulae and applications. In *Information Security, 5th International Conference, ISC 2002 Sao Paulo, Brazil, September 30 - October 2, 2002, Proceedings*, pages 272–288, 2002. 103

[BSZ05] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In Alfred Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 136–153, San Francisco, CA, USA, February 14–18, 2005. Springer, Heidelberg, Germany. 16, 106

[BV98] Dan Boneh and Ramarathnam Venkatesan. Breaking RSA may not be equivalent to factoring. In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 59–71, Espoo, Finland, May 31 – June 4, 1998. Springer, Heidelberg, Germany. 23

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science*, pages 136–145, Las Vegas, Nevada, USA, October 14–17, 2001. IEEE Computer Society Press. 122

[CD98] Ronald Cramer and Ivan Damgård. Zero-knowledge proofs for finite field arithmetic; or: Can zero-knowledge be for free? In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 424–441, Santa Barbara, CA, USA, August 23–27, 1998. Springer, Heidelberg, Germany. 36

[CDH12] Jan Camenisch, Maria Dubovitskaya, and Kristiyan Haralambiev. Efficient structure-preserving signature scheme from standard assumptions. In Ivan Visconti and Roberto De Prisco, editors, *SCN 12: 8th International Conference on Security in Communication Networks*, volume 7485 of *Lecture Notes in Computer Science*, pages 76–94, Amalfi, Italy, September 5–7, 2012. Springer, Heidelberg, Germany. 8, 40

[CDHK15] Jan Camenisch, Maria Dubovitskaya, Kristiyan Haralambiev, and Markulf Kohlweiss. Composable and modular anonymous credentials: Definitions and practical constructions. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 262–288, Auckland, New Zealand, November 30 – December 3, 2015. Springer, Heidelberg, Germany. 12, 16, 106, 109, 121, 122, 123

[CDM00] Ronald Cramer, Ivan Damgård, and Philip D. MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In Hideki Imai and Yuliang Zheng, editors, *PKC 2000: 3rd International Workshop on Theory and Practice in Public Key Cryptography*, volume 1751 of *Lecture Notes in Computer Science*, pages 354–372, Melbourne, Victoria, Australia, January 18–20, 2000. Springer, Heidelberg, Germany. 36, 110, 111, 122

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo Desmedt, editor, *Advances in Cryptology – CRYPTO'94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187, Santa Barbara, CA, USA, August 21–25, 1994. Springer, Heidelberg, Germany. 35, 103

[CF05] Henri Cohen and Gerhard Frey, editors. *Handbook of elliptic and hyperelliptic curve cryptography*. CRC Press, 2005. 4

[CF13] Dario Catalano and Dario Fiore. Vector commitments and their applications. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013: 16th International Conference on Theory and Practice of Public Key Cryptography*, volume 7778 of *Lecture Notes in Computer Science*, pages 55–72, Nara, Japan, February 26 – March 1, 2013. Springer, Heidelberg, Germany. 12

[CFL+16] Jung Hee Cheon, Pierre-Alain Fouque, Changmin Lee, Brice Minaud, and Hansol Ryu. Cryptanalysis of the new CLT multilinear map over the integers. Cryptology ePrint Archive, Report 2016/135, 2016. `http://eprint.iacr.org/2016/135`. 5

[CFW12] Dario Catalano, Dario Fiore, and Bogdan Warinschi. Efficient network coding signatures in the standard model. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012: 15th International Conference on Theory and Practice of Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 680–696, Darmstadt, Germany, May 21–23, 2012. Springer, Heidelberg, Germany. 9

[CG12] Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. *ACM Trans. Inf. Syst. Secur.*, 15(1):4, 2012. 121

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th Annual ACM Symposium on Theory of Computing*, pages 209–218, Dallas, Texas, USA, May 23–26, 1998. ACM Press. 23, 24

[CGKO06] Reza Curtmola, Juan A. Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: improved definitions and efficient constructions. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06: 13th Conference on Computer and Communications Security*, pages 79–88, Alexandria, Virginia, USA, October 30 – November 3, 2006. ACM Press. 3

[Cha81] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88, 1981. 2

[Cha82] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology – CRYPTO'82*, pages 199–203, Santa Barbara, CA, USA, 1982. Plenum Press, New York, USA. 2, 5, 10, 61

[Cha83] David Chaum. Blind signature system. In David Chaum, editor, *Advances in Cryptology – CRYPTO'83*, page 153, Santa Barbara, CA, USA, 1983. Plenum Press, New York, USA. 10

[Cha85] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, 1985. 2, 5

[Cha86] David Chaum. Showing credentials without identification: Signatures transferred between unconditionally unlinkable pseudonyms. In Franz Pichler, editor, *Advances in Cryptology – EUROCRYPT'85*, volume 219 of *Lecture Notes in Computer Science*, pages 241–244, Linz, Austria, April 1986. Springer, Heidelberg, Germany. 2

[CKL+14] Jan Camenisch, Stephan Krenn, Anja Lehmann, Gert Læssøe Mikkelsen, Gregory Neven, and Michael Østergaard Pedersen. Formal treatment of privacy-enhancing credential systems. Cryptology ePrint Archive, Report 2014/708, 2014. `http://eprint.iacr.org/2014/708`. 12, 16, 106, 122

[CKLM12] Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Sarah Meiklejohn. Malleable proof systems and applications. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 281–300, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg, Germany. 10

[CKLM13] Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Sarah Meiklejohn. Malleable signatures: Complex unary transformations and delegatable anonymous credentials. Cryptology ePrint Archive, Report 2013/179, 2013. http://eprint.iacr.org/2013/179. 9, 10

[CKW05] Jan Camenisch, Maciej Koprowski, and Bogdan Warinschi. Efficient blind signatures without random oracles. In Carlo Blundo and Stelvio Cimato, editors, *SCN 04: 4th International Conference on Security in Communication Networks*, volume 3352 of *Lecture Notes in Computer Science*, pages 134–148, Amalfi, Italy, September 8–10, 2005. Springer, Heidelberg, Germany. 10, 63

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for nontransferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118, Innsbruck, Austria, May 6–10, 2001. Springer, Heidelberg, Germany. 2, 6, 109, 121

[CL03] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN 02: 3rd International Conference on Security in Communication Networks*, volume 2576 of *Lecture Notes in Computer Science*, pages 268–289, Amalfi, Italy, September 12–13, 2003. Springer, Heidelberg, Germany. 12, 102, 105, 109, 121, 122, 123

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 56–72, Santa Barbara, CA, USA, August 15–19, 2004. Springer, Heidelberg, Germany. 12, 102, 105, 109, 121, 122, 123

[CL06] Melissa Chase and Anna Lysyanskaya. On signatures of knowledge. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 78–96, Santa Barbara, CA, USA, August 20–24, 2006. Springer, Heidelberg, Germany. 39, 126

[CL11] Sébastien Canard and Roch Lescuyer. Anonymous credentials from (indexed) aggregate signatures. In *DIM'11, Proceedings of the 2013 ACM Workshop on Digital Identity Management, Chicago, IL, USA - October 21, 2011*, pages 53–62, 2011. 109, 121, 123

[CL13] Sébastien Canard and Roch Lescuyer. Protecting privacy by sanitizing personal data: a new approach to anonymous credentials. In Kefei Chen, Qi Xie, Weidong Qiu, Ninghui Li, and Wen-Guey Tzeng,

editors, *ASIACCS 13: 8th ACM Symposium on Information, Computer and Communications Security*, pages 381–392, Hangzhou, China, May 8–10, 2013. ACM Press. 109, 121, 122, 123

[CLLT15] Jean-Sebastien Coron, Moon Sung Lee, Tancrede Lepoint, and Mehdi Tibouchi. Cryptanalysis of GGH15 multilinear maps. Cryptology ePrint Archive, Report 2015/1037, 2015. `http://eprint.iacr.org/2015/1037`. 5

[CLR15] Jung Hee Cheon, Changmin Lee, and Hansol Ryu. Cryptanalysis of the new CLT multilinear maps. Cryptology ePrint Archive, Report 2015/934, 2015. `http://eprint.iacr.org/2015/934`. 5

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 476–493, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany. 5

[CLT14] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Scale-invariant fully homomorphic encryption over the integers. Cryptology ePrint Archive, Report 2014/032, 2014. `http://eprint.iacr.org/2014/032`. 5

[CLT15] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 267–286, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany. 5

[CM99] Jan Camenisch and Markus Michels. Proving in zero-knowledge that a number is the product of two safe primes. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 107–122, Prague, Czech Republic, May 2–6, 1999. Springer, Heidelberg, Germany. 103

[CM11] Sanjit Chatterjee and Alfred Menezes. On cryptographic protocols employing asymmetric pairings - the role of Ψ revisited. *Discrete Applied Mathematics*, 159(13):1311–1322, 2011. 25, 26, 28

[CMSW14] Theresa Calderon, Sarah Meiklejohn, Hovav Shacham, and Brent Waters. Rethinking verifiably encrypted signatures: A gap in functionality and potential solutions. In Josh Benaloh, editor, *Topics in Cryptology – CT-RSA 2014*, volume 8366 of *Lecture Notes in Computer Science*, pages 349–366, San Francisco, CA, USA, February 25–28, 2014. Springer, Heidelberg, Germany. 13, 17, 80, 82, 83, 84, 92

[Cor02] Jean-Sébastien Coron. Optimal security proofs for PSS and other signature schemes. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 272–287, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer, Heidelberg, Germany. 23, 56, 57, 58

[CP93] David Chaum and Torben P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume 740 of *Lecture Notes in Computer Science*, pages 89–105, Santa Barbara, CA, USA, August 16–20, 1993. Springer, Heidelberg, Germany. 9

[CS97] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups (extended abstract). In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424, Santa Barbara, CA, USA, August 17–21, 1997. Springer, Heidelberg, Germany. 33, 39

[CV02] Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. In Vijayalakshmi Atluri, editor, *ACM CCS 02: 9th Conference on Computer and Communications Security*, pages 21–30, Washington D.C., USA, November 18–22, 2002. ACM Press. 6, 127

[Dam88] Ivan Damgård. Collision free hash functions and public key signature schemes. In David Chaum and Wyn L. Price, editors, *Advances in Cryptology – EUROCRYPT'87*, volume 304 of *Lecture Notes in Computer Science*, pages 203–216, Amsterdam, The Netherlands, April 13–15, 1988. Springer, Heidelberg, Germany. 21

[Dam90] Ivan Damgård. On the existence of bit commitment schemes and zero-knowledge proofs. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 17–27, Santa Barbara, CA, USA, August 20–24, 1990. Springer, Heidelberg, Germany. 36

[Dam00] Ivan Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 418–430, Bruges, Belgium, May 14–18, 2000. Springer, Heidelberg, Germany. 37, 121

[Dam10] Ivan Damgård. On Σ-Protocols, 2010. Manuscript v2.0. 19

[Den02] Alexander W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In Yuliang Zheng, editor,

Advances in Cryptology – ASIACRYPT 2002, volume 2501 of *Lecture Notes in Computer Science*, pages 100–109, Queenstown, New Zealand, December 1–5, 2002. Springer, Heidelberg, Germany. 24

[DFKP15] Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. Proofs of space. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 585–605, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany. 2

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. 20

[DHPS15] David Derler, Christian Hanser, Henrich C. Pöhls, and Daniel Slamanig. Towards authenticity and privacy preserving accountable workflows. In *Privacy and Identity Management for the Future Internet in the Age of Globalisation*, 2015. in press. 17

[DHS14a] David Derler, Christian Hanser, and Daniel Slamanig. Blank digital signatures: Optimization and practical experiences. In *Privacy and Identity Management for the Future Internet in the Age of Globalisation - 9th IFIP WG 9.2, 9.5, 9.6/11.7, 11.4, 11.6/SIG 9.2.2 International Summer School, Patras, Greece, September 7-12, 2014, Revised Selected Papers*, pages 201–215, 2014. 17

[DHS14b] David Derler, Christian Hanser, and Daniel Slamanig. Privacy-enhancing proxy signatures from non-interactive anonymous credentials. In *Data and Applications Security and Privacy XXVIII - 28th Annual IFIP WG 11.3 Working Conference, DBSec 2014, Vienna, Austria, July 14-16, 2014. Proceedings*, pages 49–65, 2014. 17

[DHS15a] David Derler, Christian Hanser, and Daniel Slamanig. A new approach to efficient revocable attribute-based anonymous credentials. In Jens Groth, editor, *15th IMA International Conference on Cryptography and Coding*, volume 9496 of *Lecture Notes in Computer Science*, pages 57–74, Oxford, UK, December 15–17, 2015. Springer, Heidelberg, Germany. 17

[DHS15b] 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*, volume 9048 of *Lecture Notes in Computer Science*, pages 127–144, San Francisco, CA, USA, April 20–24, 2015. Springer, Heidelberg, Germany. 12, 17, 19

[DR08] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), August 2008. 2

[FHS14] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. EUF-CMA-secure structure-preserving signatures on equivalence classes. Cryptology ePrint Archive, Report 2014/944, 2014. `http://eprint.iacr.org/2014/944`. 7, 14, 41, 47, 74, 129

[FHS15a] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 233–253, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany. 7, 8, 14, 19, 26, 39, 41, 49, 61, 62, 101, 102, 135

[FHS15b] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model. Cryptology ePrint Archive, Report 2015/626, 2015. `http://eprint.iacr.org/2015/626`. 14, 19, 41, 62

[FHS16] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Structure-preserving signatures on equivalence classes and their application to anonymous credentials. *Journal of Cryptology*, 2016. submitted. 7, 14, 19, 41, 93, 102

[Fis02] Marc Fischlin. On the impossibility of constructing non-interactive statistically-secret protocols from any trapdoor one-way function. In Bart Preneel, editor, *Topics in Cryptology – CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 79–95, San Jose, CA, USA, February 18–22, 2002. Springer, Heidelberg, Germany. 23

[Fis06] Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 60–77, Santa Barbara, CA, USA, August 20–24, 2006. Springer, Heidelberg, Germany. 10, 61

[FO98] Eiichiro Fujisaki and Tatsuaki Okamoto. A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 32–46, Espoo, Finland, May 31 – June 4, 1998. Springer, Heidelberg, Germany. 12

[Fre12] David Mandell Freeman. Improved security for linearly homomorphic signatures: A generic framework. In Marc Fischlin, Johannes

Buchmann, and Mark Manulis, editors, *PKC 2012: 15th International Conference on Theory and Practice of Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 697–714, Darmstadt, Germany, May 21–23, 2012. Springer, Heidelberg, Germany. 9

[FS87] 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*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer, Heidelberg, Germany. 35, 39

[FS90] Uriel Feige and Adi Shamir. Zero knowledge proofs of knowledge in two rounds. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 526–544, Santa Barbara, CA, USA, August 20–24, 1990. Springer, Heidelberg, Germany. 36

[FS10] Marc Fischlin and Dominique Schröder. On the impossibility of three-move blind signature schemes. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 197–215, French Riviera, May 30 – June 3, 2010. Springer, Heidelberg, Germany. 11, 19, 23, 63, 74

[Fuc09] Georg Fuchsbauer. Automorphic signatures in bilinear groups and an application to round-optimal blind signatures. Cryptology ePrint Archive, Report 2009/320, 2009. `http://eprint.iacr.org/2009/320`. 8, 40

[Fuc11] Georg Fuchsbauer. Commuting signatures and verifiable encryption. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 224–245, Tallinn, Estonia, May 15–19, 2011. Springer, Heidelberg, Germany. 12, 13

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 169–178, Bethesda, Maryland, USA, May 31 – June 2, 2009. ACM Press. 3

[GG14] Sanjam Garg and Divya Gupta. Efficient round optimal blind signatures. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 477–495, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany. 11, 62, 74

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 1–17, Athens, Greece, May 26–30, 2013. Springer, Heidelberg, Germany. 5

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual Symposium on Foundations of Computer Science*, pages 40–49, Berkeley, CA, USA, October 26–29, 2013. IEEE Computer Society Press. 5

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 498–527, Warsaw, Poland, March 23–25, 2015. Springer, Heidelberg, Germany. 5

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th Annual Symposium on Foundations of Computer Science*, pages 464–479, Singer Island, Florida, October 24–26, 1984. IEEE Computer Society Press. 20

[GH13a] Dan Garber and Elad Hazan. Playing non-linear games with linear oracles. In *54th Annual Symposium on Foundations of Computer Science*, pages 420–428, Berkeley, CA, USA, October 26–29, 2013. IEEE Computer Society Press. 5

[GH13b] Gerwin Gsenger and Christian Hanser. Improving the efficiency of elliptic curve scalar multiplication using binary huff curves. In *Security Engineering and Intelligence Informatics - CD-ARES 2013 Workshops: MoCrySEn and SeCIHD, Regensburg, Germany, September 2-6, 2013. Proceedings*, pages 155–167, 2013. 17

[Gha16] Essam Ghadafi. Short structure-preserving signatures. In Kazue Sako, editor, *Topics in Cryptology – CT-RSA 2016*, volume 9610 of *Lecture Notes in Computer Science*, pages 305–321, San Francisco, CA, USA, February 29 – March 4, 2016. Springer, Heidelberg, Germany. 40

[GHMS14] Craig Gentry, Shai Halevi, Hemanta K. Maji, and Amit Sahai. Zeroizing without zeroes: Cryptanalyzing multilinear maps without encodings of zero. Cryptology ePrint Archive, Report 2014/929, 2014. `http://eprint.iacr.org/2014/929`. 5

[GK16] Shafi Goldwasser and Yael Tauman Kalai. Cryptographic assumptions: A position paper. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 505–522, Tel Aviv, Israel, January 10–13, 2016. Springer, Heidelberg, Germany. 22

[GKM+00] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *41st Annual Symposium on Foundations of Computer Science*, pages 325–335, Redondo Beach, California, USA, November 12–14, 2000. IEEE Computer Society Press. 23

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st Annual ACM Symposium on Theory of Computing*, pages 25–32, Seattle, Washington, USA, May 15–17, 1989. ACM Press. 20, 21, 92

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th Annual ACM Symposium on Theory of Computing*, pages 291–304, Providence, Rhode Island, USA, May 6-8, 1985. ACM Press. 32

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988. 20, 38

[GMR01] Yael Gertner, Tal Malkin, and Omer Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *42nd Annual Symposium on Foundations of Computer Science*, pages 126–135, Las Vegas, Nevada, USA, October 14–17, 2001. IEEE Computer Society Press. 23

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to prove all NP-statements in zero-knowledge, and a methodology of cryptographic protocol design. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 171–185, Santa Barbara, CA, USA, August 1987. Springer, Heidelberg, Germany. 32

[GO96] Oded Goldreich and Rafail Ostrovsky. Software protection and simulation on oblivious rams. *J. ACM*, 43(3):431–473, 1996. 3

[Gol01] Oded Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, Cambridge, UK, 2001. 92

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06: 13th Conference on Computer and Communications Security*, pages 89–98, Alexandria, Virginia, USA, October 30 – November 3, 2006. ACM Press. Available as Cryptology ePrint Archive Report 2006/309. 4

[GR04] Craig Gentry and Zulfikar Ramzan. Eliminating random permutation oracles in the Even-Mansour cipher. In Pil Joong Lee, editor, *Advances in Cryptology – ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 32–47, Jeju Island, Korea, December 5–9, 2004. Springer, Heidelberg, Germany. 23

[Gro10] Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 321–340, Singapore, December 5–9, 2010. Springer, Heidelberg, Germany. 12

[Gro15] Jens Groth. Efficient fully structure-preserving signatures for large messages. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015, Part I*, volume 9452 of *Lecture Notes in Computer Science*, pages 239–259, Auckland, New Zealand, November 30 – December 3, 2015. Springer, Heidelberg, Germany. 8

[GRS+11] Sanjam Garg, Vanishree Rao, Amit Sahai, Dominique Schröder, and Dominique Unruh. Round optimal blind signatures. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 630–648, Santa Barbara, CA, USA, August 14–18, 2011. Springer, Heidelberg, Germany. 11, 62

[GS07] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. Cryptology ePrint Archive, Report 2007/155, 2007. `http://eprint.iacr.org/2007/155`. 5

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432, Istanbul, Turkey, April 13–17, 2008. Springer, Heidelberg, Germany. 5, 8, 10, 39, 46

[GS12] Essam Ghadafi and Nigel P. Smart. Efficient two-move blind signatures in the common reference string model. In Dieter Gollmann and Felix C. Freiling, editors, *ISC 2012: 15th International Conference on Information Security*, volume 7483 of *Lecture Notes*

in Computer Science, pages 274–289, Passau, Germany, September 19–21, 2012. Springer, Heidelberg, Germany. 10

[GT00] Rosario Gennaro and Luca Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *41st Annual Symposium on Foundations of Computer Science*, pages 305–313, Redondo Beach, California, USA, November 12–14, 2000. IEEE Computer Society Press. 23

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd Annual ACM Symposium on Theory of Computing*, pages 99–108, San Jose, California, USA, June 6–8, 2011. ACM Press. 22

[Hes04] Florian Hess. On the security of the verifiably-encrypted signature scheme of Boneh, Gentry, Lynn and Shacham. *Information Processing Letters*, 89(3):111–114, 2004. 13, 17

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. 20

[HKKL07] Carmit Hazay, Jonathan Katz, Chiu-Yuen Koo, and Yehuda Lindell. Concurrently-secure blind signatures without random oracles or setup assumptions. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 323–341, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Heidelberg, Germany. 10, 11, 61

[HMV03] Darrel Hankerson, Alfred J. Menezes, and Scott Vanstone. *Guide to Elliptic Curve Cryptography*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003. 4

[HR] Christian Hanser and Sebastian Ramacher. ECCelerate Java Toolkit. `https://jce.iaik.tugraz.at/`. 17

[HRS15] Christian Hanser, Max Rabkin, and Dominique Schröder. Verifiably encrypted signatures: Security revisited and a new construction. In Günther Pernul, Peter Y. A. Ryan, and Edgar R. Weippl, editors, *ESORICS 2015: 20th European Symposium on Research in Computer Security, Part I*, volume 9326 of *Lecture Notes in Computer Science*, pages 146–164, Vienna, Austria, September 21–25, 2015. Springer, Heidelberg, Germany. 7, 14, 19, 80

[HS13a] Christian Hanser and Daniel Slamanig. Blank digital signatures. In Kefei Chen, Qi Xie, Weidong Qiu, Ninghui Li, and Wen-Guey

Tzeng, editors, *ASIACCS 13: 8th ACM Symposium on Information, Computer and Communications Security*, pages 95–106, Hangzhou, China, May 8–10, 2013. ACM Press. 17

[HS13b] Christian Hanser and Daniel Slamanig. Efficient simultaneous privately and publicly verifiable robust provable data possession from elliptic curves. In *SECRYPT 2013 - Proceedings of the 10th International Conference on Security and Cryptography, Reykjavík, Iceland, 29-31 July, 2013*, pages 15–26, 2013. 17

[HS13c] Christian Hanser and Daniel Slamanig. Warrant-hiding delegation-by-certificate proxy signature schemes. In Goutam Paul and Serge Vaudenay, editors, *Progress in Cryptology - INDOCRYPT 2013: 14th International Conference in Cryptology in India*, volume 8250 of *Lecture Notes in Computer Science*, pages 60–77, Mumbai, India, December 7–10, 2013. Springer, Heidelberg, Germany. 17

[HS14] Christian Hanser and Daniel Slamanig. Structure-preserving signatures on equivalence classes and their application to anonymous credentials. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 491–511, Kaoshiung, Taiwan, R.O.C., December 7–11, 2014. Springer, Heidelberg, Germany. 7, 8, 14, 39, 41, 93, 95, 102

[HW13] Christian Hanser and Christian Wagner. Speeding up the fixed-base comb method for faster scalar multiplication on koblitz curves. In *Security Engineering and Intelligence Informatics - CD-ARES 2013 Workshops: MoCrySEn and SeCIHD, Regensburg, Germany, September 2-6, 2013. Proceedings*, pages 168–179, 2013. 17

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *30th Annual Symposium on Foundations of Computer Science*, pages 230–235, Research Triangle Park, North Carolina, October 30 – November 1, 1989. IEEE Computer Society Press. 20

[ILV11] Malika Izabachène, Benoît Libert, and Damien Vergnaud. Block-wise P-signatures and non-interactive anonymous credentials with efficient attributes. In Liqun Chen, editor, *13th IMA International Conference on Cryptography and Coding*, volume 7089 of *Lecture Notes in Computer Science*, pages 431–450, Oxford, UK, December 12–15, 2011. Springer, Heidelberg, Germany. 12

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st Annual ACM Symposium on Theory of Computing*, pages 44–61, Seattle, Washington, USA, May 15–17, 1989. ACM Press. 20

[IR90] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In Shafi Goldwasser, editor, *Advances in Cryptology – CRYPTO'88*, volume 403 of *Lecture Notes in Computer Science*, pages 8–26, Santa Barbara, CA, USA, August 21–25, 1990. Springer, Heidelberg, Germany. 23

[JLO97] Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (extended abstract). In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 150–164, Santa Barbara, CA, USA, August 17–21, 1997. Springer, Heidelberg, Germany. 10, 63

[JMSW02] Robert Johnson, David Molnar, Dawn Xiaodong Song, and David Wagner. Homomorphic signature schemes. In Bart Preneel, editor, *Topics in Cryptology – CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 244–262, San Jose, CA, USA, February 18–22, 2002. Springer, Heidelberg, Germany. 9

[Jou00] Antoine Joux. A one round protocol for tripartite diffie-hellman. In *Algorithmic Number Theory, 4th International Symposium, ANTS-IV, Leiden, The Netherlands, July 2-7, 2000, Proceedings*, pages 385–394, 2000. 4

[Kat10] Jonathan Katz. *Digital Signatures*. Springer, 2010. 19, 39

[KL15] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2 edition, 2015. 19

[KO97] Eyal Kushilevitz and Rafail Ostrovsky. Replication is NOT needed: SINGLE database, computationally-private information retrieval. In *38th Annual Symposium on Foundations of Computer Science*, pages 364–373, Miami Beach, Florida, October 19–22, 1997. IEEE Computer Society Press. 3

[Kob89] Neal Koblitz. Hyperelliptic cryptosystems. *Journal of Cryptology*, 1(3):139–150, 1989. 4

[KPW15] Eike Kiltz, Jiaxin Pan, and Hoeteck Wee. Structure-preserving signatures from standard assumptions, revisited. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 275–295, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany. 8, 40

[KST99] Jeong Han Kim, Daniel R. Simon, and Prasad Tetali. Limits on the efficiency of one-way permutation-based hash functions. In *40th Annual Symposium on Foundations of Computer Science*, pages 535–542, New York, New York, USA, October 17–19, 1999. IEEE Computer Society Press. 23

[KZ06] Aggelos Kiayias and Hong-Sheng Zhou. Concurrent blind signatures without random oracles. In Roberto De Prisco and Moti Yung, editors, *SCN 06: 5th International Conference on Security in Communication Networks*, volume 4116 of *Lecture Notes in Computer Science*, pages 49–62, Maiori, Italy, September 6–8, 2006. Springer, Heidelberg, Germany. 61

[KZG10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 177–194, Singapore, December 5–9, 2010. Springer, Heidelberg, Germany. 12, 93

[Lam79] Leslie Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, October 1979. 20

[Lin03] Yehuda Lindell. Bounded-concurrent secure two-party computation without setup assumptions. In *35th Annual ACM Symposium on Theory of Computing*, pages 683–692, San Diego, California, USA, June 9–11, 2003. ACM Press. 10, 63

[Lip12] Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 169–189, Taormina, Sicily, Italy, March 19–21, 2012. Springer, Heidelberg, Germany. 12

[LOS+06] Steve Lu, Rafail Ostrovsky, Amit Sahai, Hovav Shacham, and Brent Waters. Sequential aggregate signatures and multisignatures without random oracles. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 465–485, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany. 13

[LPJY13] Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Linearly homomorphic structure-preserving signatures and their applications. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 289–307, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany. 8, 9, 39

[LPY15] Benoît Libert, Thomas Peters, and Moti Yung. Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015,*

Part II, volume 9216 of *Lecture Notes in Computer Science*, pages 296–316, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany. 8, 40

[LRSW99] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In Howard M. Heys and Carlisle M. Adams, editors, *SAC 1999: 6th Annual International Workshop on Selected Areas in Cryptography*, volume 1758 of *Lecture Notes in Computer Science*, pages 184–199, Kingston, Ontario, Canada, August 9–10, 1999. Springer, Heidelberg, Germany. 2, 6, 10

[LSS14] Adeline Langlois, Damien Stehlé, and Ron Steinfeld. GGHLite: More efficient multilinear maps from ideal lattices. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 239–256, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany. 5

[Mau05] Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *Lecture Notes in Computer Science*, pages 1–12, Cirencester, UK, December 19–21, 2005. Springer, Heidelberg, Germany. 24

[Mer88] Ralph C. Merkle. A digital signature based on a conventional encryption function. In Carl Pomerance, editor, *Advances in Cryptology – CRYPTO'87*, volume 293 of *Lecture Notes in Computer Science*, pages 369–378, Santa Barbara, CA, USA, August 16–20, 1988. Springer, Heidelberg, Germany. 12

[MGGR13] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. Zerocoin: Anonymous distributed E-cash from Bitcoin. In *2013 IEEE Symposium on Security and Privacy*, pages 397–411, Berkeley, California, USA, May 19–22, 2013. IEEE Computer Society Press. 2

[Mil86a] Victor Miller. Short programs for functions on curves, 1986. unpublished manuscript. 4

[Mil86b] Victor S. Miller. Use of elliptic curves in cryptography. In Hugh C. Williams, editor, *Advances in Cryptology – CRYPTO'85*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426, Santa Barbara, CA, USA, August 18–22, 1986. Springer, Heidelberg, Germany. 4

[MNT01] A. Miyaji, M. Nakabayashi, and S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Transactions on Fundamentals*, E84-A (5):1234–1243, 2001. 4

[MRK03] Silvio Micali, Michael O. Rabin, and Joe Kilian. Zero-knowledge sets. In *44th Annual Symposium on Foundations of Computer Science*, pages 80–91, Cambridge, Massachusetts, USA, October 11–14, 2003. IEEE Computer Society Press. 12

[MSF10] Sarah Meiklejohn, Hovav Shacham, and David Mandell Freeman. Limitations on transformations from composite-order to prime-order groups: The case of round-optimal blind signatures. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 519–538, Singapore, December 5–9, 2010. Springer, Heidelberg, Germany. 10

[MSK02] Shigeo Mitsunari, Ryuichi Sakai, and Masao Kasahara. A new traitor tracing. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 85(2):481–484, 2002. 26

[Nak09] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2009. 2

[Nao03] Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 96–109, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Heidelberg, Germany. 22

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd Annual ACM Symposium on Theory of Computing*, pages 427–437, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press. 29

[Oka93] Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53, Santa Barbara, CA, USA, August 16–20, 1993. Springer, Heidelberg, Germany. 10

[Oka06] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 80–99, New York, NY, USA, March 4–7, 2006. Springer, Heidelberg, Germany. 63

[Ped92] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140, Santa Barbara, CA, USA, August 11–15, 1992. Springer, Heidelberg, Germany. 12, 30, 31

[Pol93] J.M. Pollard. The lattice sieve. In ArjenK. Lenstra and Jr. Lenstra, HendrikW., editors, *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*, pages 43–49. Springer Berlin Heidelberg, 1993. 4

[PPK+15] Sunoo Park, Krzysztof Pietrzak, Albert Kwon, Joël Alwen, Georg Fuchsbauer, and Peter Gaži. Spacemint: A cryptocurrency based on proofs of space. Cryptology ePrint Archive, Report 2015/528, 2015. `http://eprint.iacr.org/2015/528`. 2

[PS15a] Omer Paneth and Amit Sahai. On the equivalence of obfuscation and multilinear maps. Cryptology ePrint Archive, Report 2015/791, 2015. `http://eprint.iacr.org/2015/791`. 5

[PS15b] David Pointcheval and Olivier Sanders. Short randomizable signatures. Cryptology ePrint Archive, Report 2015/525, 2015. `http://eprint.iacr.org/2015/525`. 12

[PV06] Pascal Paillier and Jorge L. Villar. Trading one-wayness against chosen-ciphertext security in factoring-based encryption. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology – ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 252–266, Shanghai, China, December 3–7, 2006. Springer, Heidelberg, Germany. 23

[PWH+13] Klaus Potzmader, Johannes Winter, Daniel Hein, Christian Hanser, Peter Teufl, and Liqun Chen. Group signatures on mobile devices: Practical experiences. In *Trust and Trustworthy Computing - 6th International Conference, TRUST 2013, London, UK, June 17-19, 2013. Proceedings*, pages 47–64, 2013. 17

[Rab05] Michael O. Rabin. How to exchange secrets with oblivious transfer. Cryptology ePrint Archive, Report 2005/187, 2005. `http://eprint.iacr.org/2005/187`. 2

[RAD78] R L Rivest, L Adleman, and M L Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation, Academia Press*, pages 169–179, 1978. 3

[RGK05] Mario Di Raimondo, Rosario Gennaro, and Hugo Krawczyk. Secure off-the-record messaging. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, WPES 2005, Alexandria, VA, USA, November 7, 2005*, pages 81–89, 2005. 2

[Rog06] Phillip Rogaway. Formalizing human ignorance. In Phong Q. Nguyen, editor, *Progress in Cryptology - VIETCRYPT 06: 1st International Conference on Cryptology in Vietnam*, volume 4341 of *Lecture Notes in Computer Science*, pages 211–228, Hanoi, Vietnam, September 25–28, 2006. Springer, Heidelberg, Germany. 21

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd Annual ACM Symposium on Theory of Computing*, pages 387–394, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press. 20

[RS92] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444, Santa Barbara, CA, USA, August 11–15, 1992. Springer, Heidelberg, Germany. 29

[RS09] Markus Rückert and Dominique Schröder. Security of verifiably encrypted signatures and a construction without random oracles. In Hovav Shacham and Brent Waters, editors, *PAIRING 2009: 3rd International Conference on Pairing-based Cryptography*, volume 5671 of *Lecture Notes in Computer Science*, pages 17–34, Palo Alto, CA, USA, August 12–14, 2009. Springer, Heidelberg, Germany. 13, 17, 82, 83, 84

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978. 4, 20

[RSG98] Michael G. Reed, Paul F. Syverson, and David M. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, 1998. 2

[Rüc09] Markus Rückert. Verifiably encrypted signatures from RSA without NIZKs. In Bimal K. Roy and Nicolas Sendrier, editors, *Progress in Cryptology - INDOCRYPT 2009: 10th International Conference in Cryptology in India*, volume 5922 of *Lecture Notes in Computer Science*, pages 363–377, New Delhi, India, December 13–16, 2009. Springer, Heidelberg, Germany. 13

[Rüc10] Markus Rückert. Lattice-based blind signatures. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 413–430, Singapore, December 5–9, 2010. Springer, Heidelberg, Germany. 10

[Rud92] Steven Rudich. The use of interaction in public cryptosystems (extended abstract). In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 242–251, Santa Barbara, CA, USA, August 11–15, 1992. Springer, Heidelberg, Germany. 23

[SBZ02] Ron Steinfeld, Laurence Bull, and Yuliang Zheng. Content extraction signatures. In Kwangjo Kim, editor, *ICISC 01: 4th International Conference on Information Security and Cryptology*, volume

2288 of *Lecture Notes in Computer Science*, pages 285–304, Seoul, Korea, December 6–7, 2002. Springer, Heidelberg, Germany. 9

[SC12] Jae Hong Seo and Jung Hee Cheon. Beyond the limitation of prime-order bilinear groups, and round optimal blind signatures. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 133–150, Taormina, Sicily, Italy, March 19–21, 2012. Springer, Heidelberg, Germany. 10

[Sch80] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980. 135, 137

[Sch90] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252, Santa Barbara, CA, USA, August 20–24, 1990. Springer, Heidelberg, Germany. 34, 35, 39

[Sch15] Berry Schoenmakers. Lecture Notes Cryptographic Protocols, 2015. Manuscript v1.2. 19, 35

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology – CRYPTO'84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53, Santa Barbara, CA, USA, August 19–23, 1984. Springer, Heidelberg, Germany. 4

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266, Konstanz, Germany, May 11–15, 1997. Springer, Heidelberg, Germany. 24

[Sil86] Joseph H. Silverman. *The arithmetic of elliptic curves*. Graduate texts in mathematics. Springer, New York, Berlin, 1986. 2e tirage corrig 1992. 4

[Sim98] Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 334–345, Espoo, Finland, May 31 – June 4, 1998. Springer, Heidelberg, Germany. 23

[SNF11] Amang Sudarsono, Toru Nakanishi, and Nobuo Funabiki. Efficient proofs of attributes in pairing-based anonymous credential system. In Simone Fischer-Hübner and Nicholas Hopper, editors, *Privacy Enhancing Technologies - 11th International Symposium, PETS*,

pages 246–263, Waterloo, ON, Canada, July 27–29, 2011. Springer, Berlin, Germany. 121

[Ver01] Eric R. Verheul. Self-blindable credential certificates from the Weil pairing. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 533–551, Gold Coast, Australia, December 9–13, 2001. Springer, Heidelberg, Germany. 9

[Ver10] Frederik Vercauteren. Optimal pairings. *IEEE Transactions on Information Theory*, 56(1):455–461, 2010. 4, 5

[Wat05] Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127, Aarhus, Denmark, May 22–26, 2005. Springer, Heidelberg, Germany. 8

[Wei40] André Weil. Sur les fonctions algébriques à corps de constantes fini. *C. R. Acad. Sci. Paris*, 210:592–594, 1940. 4

[Wil95] Andrew Wiles. Modular elliptic curves and fermat's last theorem. *Annals of Mathematics*, 141(3):443–551, 1995. 4

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposium on on Symbolic and Algebraic Computation*, EUROSAM'79, pages 216–226, London, UK, 1979. Springer-Verlag. 135, 137

[ZSS03] Fangguo Zhang, Reihaneh Safavi-Naini, and Willy Susilo. Efficient verifiably encrypted signature and partially blind signature from bilinear pairings. In Thomas Johansson and Subhamoy Maitra, editors, *Progress in Cryptology - INDOCRYPT 2003: 4th International Conference in Cryptology in India*, volume 2904 of *Lecture Notes in Computer Science*, pages 191–204, New Delhi, India, December 8–10, 2003. Springer, Heidelberg, Germany. 13

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am …… ……..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…… ……..
date (signature)