Enhancing Privacy Protection: Set Membership, Range Proofs, and the Extended Access Control

THÈSE N^{O} 7112 (2017)

PRÉSENTÉE LE 11 AVRIL 2017

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE SÉCURITÉ ET DE CRYPTOGRAPHIE
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Rafik CHAABOUNI

acceptée sur proposition du jury:

Prof. A. Lenstra, président du jury
Prof. S. Vaudenay, Prof. H. Lipmaa, directeurs de thèse
Prof. J. Groth, rapporteur
Prof. A. Kiayias, rapporteur
Prof. B. A. Ford, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2017

In memory of my beloved Sister, Sonia Chaabouni-Piralla,
In memory of my beloved Father, Hamed Chaabouni,
And to my dear Mother, Amel Chaabouni.

Abstract

Privacy has recently gained an importance beyond the field of cryptography. In that regard, the main goal behind this thesis is to enhance privacy protection. All of the necessary mathematical and cryptographic preliminaries are introduced at the start of this thesis. We then show in Part I how to improve set membership and range proofs, which are cryptographic primitives enabling better privacy protection. Part II shows how to improve the standards for Machine Readable Travel Documents (MRTDs), such as biometric passports.

Regarding set membership proofs, we provide an efficient protocol based on the Boneh-Boyen signature scheme. We show that alternative signature schemes can be used and we provide a general protocol description that can be applied for any secure signature scheme. We also show that signature schemes in our design can be replaced by cryptographic accumulators. For range proofs, we provide interactive solutions where the range is divided in a base u and the u-ary digits are handled by one of our set membership proofs. A general construction is also provided for any set membership proof. We additionally explain how to handle arbitrary ranges with either two range proofs or with an improved solution based on sumset representation. These efficient solutions achieve, to date, the lowest asymptotical communication load. Furthermore, this thesis shows that the first efficient non-interactive range proof is insecure. This thesis thus provides the first efficient and secure non-interactive range proof.

In the case of MRTDs, two standards exist: one produced by the International Civil Aviation Organization (ICAO) and the other by the European Union, which is called the Extended Access Control (EAC). Although this thesis focuses on the EAC, which is supposed to solve all privacy concerns, it shows that both standards fail to provide complete privacy protection. Lastly, we provide several solutions to improve them.

Keywords: cryptography, privacy, set membership, range proof, machine readable travel document, MRTD, sumset representation, proof of knowledge, zero-knowledge, NIZK, sigma protocol, commitment, public key cryptography, digital signature, cryptographic accumulator, biometric passport, electronic passport, extended access control, EAC, terminal revocation.

Résumé

Les problèmes liés à la sphère privée ont gagné cette dernière décénie une importance qui va au-delà du domaine de la cryptographie. De ce fait, l'enjeux principale de cette thèse est d'amméliorer la protection de la sphère privée. Dans un premier temps, les prérequis mathématiques et cryptographiques seront expliqués. La première partie de cette thèse se consacre donc à l'amélioration de primitives cryptographiques liées à la sphère privée (appartenance à un ensemble, appartenance à un intervalle). La seconde partie de cette thèse se concentre sur l'amélioration des standards régissant les documents de voyage lisible à distance (MRTDs) tels que les passeports biométriques.

En ce qui concerne l'appartenance à un ensemble, notre solution se base sur les signatures numériques proposées par Boneh et Boyen. Nous montrons par la suite que notre construction peut s'adapter à d'autres signatures numériques, ainsi qu'aux accumulateurs cryptographiques. Pour ce qui est de l'appartenance à un intervalle, nous proposons des solutions interactives où l'intervalle est decomposé en base u et où les subdivisions sont résolues par des apartenances à un ensemble, notamment celles que nous élaborons. Nous expliquons aussi comment prendre en charge des intervalles arbitraires soit en combinant deux appartenances à un intervalle, soit grâce à une unique apartenance basée sur une représentation par somme d'ensembles. Jusqu'à présent, ces solutions sont les plus efficaces asymptotiquement vis à vis des communications. Nous prouvons également que le premier protocole pour une appartenance non-interactive à un intervalle est défaillant. Ainsi nous proposons la première alternative sécurisée.

Dans le cas des MRTDs, deux standards existent. Le premier est sous la responsabilité de l'Organisation de l'Aviation Civile Internationale (OACI), tandis que le second, appelé Controle d'Accès Étendu (EAC), a été mandaté par l'Union Européenne. Bien que cette thèse se concentre à l'étude de l'EAC, nous fournissons une analyse de la sécurité liée à la sphère privée des deux standards. Des solutions pour les améliorer sont aussi fournies.

Mots Clés : cryptographie, sphère privée, appartenance à un ensemble, appartenance à un intervalle, document de voyage lisible à distance, MRTD, représentation par somme d'ensembles, preuve de connaissance, preuve à divulgation nulle, NIZK, protocole sigma, mise en gage, cryptographie asymétrique, signature numérique, accumulateur cryptographique, passeport biométrique, passeport électronique, contrôle d'accès étendu, EAC, révocation de terminaux.

Acknowledgments

My passion for cryptography started in Egypt at the early age of 11, when I was introduced to the Rosetta stone and its decryption by Champollion. Little did I know at the time how vast the field of cryptography is. Nevertheless, my course was set, sails were up and I eventually reached the door of Professor Serge Vaudenay, at the EPFL.

It is difficult to express my respect and gratitude towards Professor Serge Vaudenay in words (but I will try). First of all, I am thankful to him for testing the strength of my motivation and for accepting me as his PhD student. I am also thankful for his patience in teaching us to perform rigorous research. There is no doubt that he holds the position of father at the LASEC, with a pedagogy based on the excellency of the French Elite Schools, and it will be with great honor for me to join his group of academic sons.

Professor Helger Lipmaa, besides being my co-supervisor, has also become a good friend to whom I look up to. Like a wonderful uncle, his joyful attitude lightened my mood during the most difficult times of my life. As a researcher, I knew his work before meeting him, and I was moved when I witnessed his passion for cryptography at the Asiacrypt 2008. Helger is the personification of Serge's motto: "*La crypto, c'est rigolo*" (crypto is fun).

I thank Professor Babak Falsafi for his help and support as a mediator during the process of having a co-supervisor, as well as following my progress during the last stages of my doctoral studies. I also thank Professor Arjen Lenstra, the president of the jury, as well as the reviewers Professor Bryan Ford, Professor Jens Groth, and Professor Aggelos Kiayias. I thank my co-authors for our pleasant collaborations and cite them in alphabetical order as is the custom in cryptography: Dr. Jan Camenisch, Professor Helger Lipmaa, Professor abhi shelat[1], Professor Serge Vaudenay, and Dr. Bingsheng Zhang. This thesis has been supported by many grants, which I would also like to express my thanks for: the European Commission through the ICT program (Contract ICT-2007-216676 ECRYPT II) and through the European Regional Development Fund; the Swiss National Science Foundation (SNSF, grant 200021-124575); the Estonian Science Foundation (grant #9303); the European Social Fund's Doctoral Studies and Internationalization Programme DoRa; and the Estonian Center of Excellence in Computer Science (EXCS, research theme IUT2-1).

[1] Note that abhi shelat requires his name to be written in lower case.

Acknowledgments

I express my warm thanks and sympathy to the members of the International Telecommunication Union (ITU) and the Arab ICT Organization (AICTO), who were dear friends of my beloved father. A special thank you goes to the Secretary General of AICTO, Mme Khédija Hamouda Ghariani, who invited me twice as a guest speaker at the AICTO.

I would now like to thank my academic families, the LASEC at EPFL and the crypto group at the University of Tartu.

Although not officially a member of the LASEC, Christine Vaudenay plays the crucial role of "mother" in the lab, and deserves special thanks in that respect. While Serge is a strict "father", Christine reassures us and gives us motivation. LASEC being a large family, it absolutely requires the presence of a nanny; this responsibility was taken by Martine Corval, and without her we would have been lost children. Special thanks go as well to *il capo* Martin Vuagnoux and *il capo* Sylvain Pasini. The stories told by Martin kept us in good spirits, and playing with Sylvain was great fun. I would also like to thank Thomas Baignères in particular, for teaching me a shameful lesson on svn. Sorry, Thomas. The LASEC family is composed of wonderful people that guided me, taught me, helped me, and gave me support. I notably thank: Philippe Oechslin for his lesson on security (not to be confused with cryptography); Pascal Junod, Gildas Avoine, and Jean Monnerat for their teaching as teaching assistants; Claude Barral for teaching us how to fake fingerprints; Pouyan Sepehrdad for being a nice office mate; Khaled Ouafi with whom I started EPFL in 2002; Aslı Bay, Atefeh Mashatan, and Petr Sušil for their friendship; and Sonia Mihaela Bogos for her help. Furthermore, I had the privilege of supervising the extremely talented Alexandre Duc during a semester project, before he joined LASEC.

In the crypto group at the University of Tartu, my regards go to present and former members, notably to the eccentric Professor Dominique Unruh, the almighty Sven Laur who translated my abstract into Estonian, Bingsheng Zhang for his Chinese teaching, as well as Dan Bogdanov and Liina Kamm for their friendship.

In the University of Tartu, I would like to thank the head of the Computer Science Institute, Professor Jaak Vilo, who funded and allowed me to undertake the controversial project on drone insecurity. My thanks go as well to Maria Gaiduk and Mirjam Paales for their IT support. I also thank the group of projects coordinators and affiliates for their help in integrating me at the University of Tartu, in alphabetical order: Laura Kalda, Martin Kaljula, Mari Krusten, Kristin Liba, Eva Pruusapuu, Kadri Raudvere, Kairit Shor, and Anneli Vainumäe. I thank as well Naved Ahmed, Walid Fdhila, and Fredrik Payman Milani for bringing me joy at the university and in Tartu.

I thank my friends for making me laugh, bringing me joy, keeping me sane, and cheering me up during my most difficult moments: the gang of Florimont (Sven Gowal, Michel Sède, Lionel Coulot, Guillaume Lovey, Sébastien Jacquemoud, ...), the gang of syscom-ic (José Camacho, Mohamed Chaabouni, Loana Chatelain, Marcelo Fernandez, Florent Garcin, Julien Hamilton, Bao-Lan Huynh, ...), Kairi Kübarsepp and her family for giving me the strength to rise up from grief, and the gang of the small dwarf (Jérémy Castéra, Maud Vadon, and Vincent Perrier). I also thank Ali Adil and Kairit Pärna for hosting me so many times in Tallinn and for all of our cooking sessions. I thank Jessica Walsh for proofreading the English in my thesis. Foremost, I thank Amnir Hadachi and Nebil Mansour, on whom I can always count on for their help and support.

Last but not least, I thank my family for trusting me, giving me support and their unconditional love. I dedicate this thesis to my beloved sister, Sonia Chaabouni-Piralla, who left us on the 17th May 2011; to my beloved father, Hamed Chaabouni, who left us on the 9th May 2014; and to my dear mother, Amel Chaabouni. It is thanks to their love, knowledge, teachings and support that I was able to pursue my dreams. My gratitude goes as well to my nephews, Hedi and Samy Piralla, for their spark of curiosity, and to my brother-in-law, Patrick Piralla. I also deeply thank my grandmother, Zohra Kammoun-Sellami, who has constantly supported me throughout my doctoral studies.

Geneva, 11th January 2017 R. C.

Contents

Chapter 1

Introduction

The intuitive concept of privacy is the ability of a person to control her personal information dissemination. With the development of electronic services (e-voting, e-taxes, e-cash, ...), and with the introduction of internet social networks such as Facebook and Google+, concerns about privacy increased significantly. Privacy is of even greater concern when children are involved. In that regard, social networks have limited their access to children above 13 years old. This is further confirmed by the International Telecommunication Union (ITU), which launched the Child Online Protection (COP) Initiative [ITU15]. Furthermore, this concern about privacy is being strengthened by the threat of identity theft.

The first and main issue of interest in this thesis, targets specific cryptographic protocols called ***set membership and range proofs***, which consist of proving that a secret element belongs to a public set. These protocols are considered building blocks in cryptography and need to meet several security requirements, such as revealing absolutely no information about the element, except its membership of the public set. Set membership proofs allow users to prove that their committed secret belongs to a small public set of elements, whilst keeping their secret hidden. A typical example can be given in the case of e-voting, where users are the voters, their committed secret is their vote in their secret ballot, and the public set is, for instance, the public list of candidates in an election. By using a set membership proof, voters can show the validity of their vote without revealing the vote itself. Note that for set membership proofs, the public set is usually of a small size with no specific structure. Range proofs are a special case of set membership proofs, where the public set is an integer interval of possibly large size. Moreover, as the elements in an integer interval occur in consecutive order, special techniques can be applied to improve efficiency. Range proofs could thus be used to enforce age restrictions when combined with certified electronic identities.

The second issue of concern is the privacy relating to ***Machine Readable Travel Documents (MRTDs)*** such as biometric passports. MRTDs hold the details of our certified national identity. These are privacy sensitive and thus need to be protected accordingly. The initial standards

regarding MRTDs focused on border control surveillance security and set aside privacy. As a consequence, an unlimited right was given to terminals, the reading devices of border controls, to read the data of any MRTD. Several attempts have been made later in order to provide better privacy, notably by defining a new standard called the ***Extended Access Control (EAC)***. Unfortunately, major flaws remained due to the hardware choices for MRTDs. One of the flaws that is studied in this thesis, concerns the revocation of the rights belonging to the terminal of a border patrol. Indeed, the standards for biometric passports have no proper revocation mechanism for border patrols. This problem is referred to as the ***terminal revocation*** problem.

1.1 Motivation

The initial motivation behind *set membership and range proofs* as a cryptographic building block, came from the way cryptographers traditionally defined and constructed cryptographic protocols [Gol01, Gol04]. Informally, a cryptographic protocol is a secure protocol that implements a specific functionality. However, the definition of security is often based on an idealized model of potential adversaries. One common example is the security model that restricts the behavior of adversaries by disallowing them to actively disrupt the procedure of protocols. In this model, adversaries are only allowed to gather information from honest communications. Therefore, they are called *semi-honest*. However, in order to provide security against any type of adversaries, honest communications need to be enforced. Set membership and range proofs help enforce honest communication by providing a means of verifying that private parameters are chosen from the correct corresponding sets.

Further important applications were later found for set membership and range proofs. For instance, they have an important impact on several online services, such as in the case of e-services (electronic voting, electronic taxation, ...). For electronic voting, a set membership proof can be used to prove that a ballot is valid without revealing the value of the ballot. This is of particular interest to countries that have compulsory voting [IIDEA02], such as Brazil and Australia. In electronic taxation, range proofs could be used to attest range taxation for incomes and wealth, without leaking any other information. Whereas Switzerland has some initial projects regarding e-services [ODIHR12], some countries such as Estonia have already rolled out complete systems [EEC15], despite security flaws [SFD+14].

Set membership and range proofs also occur in the context of anonymous credentials [BNF12, GGM14]. Assume that a user is issued a credential containing a number of attributes such as nationality. Furthermore assume that the user needs to prove she is from a European country. As the list of European countries is public, the user has to show that she possesses a credential containing one of those countries as a nationality, without leaking her specific country. This can be simply achieved with the help of set membership proofs. Furthermore, range proofs applied to age can also bring a solution in the framework of the Child Online Protection (COP)

Initiative [ITU15]. For example, a user with passport credentials or with a certified electronic identity (eID) might wish to prove that her age is within some range, such as greater than 18, or between 13 and 18 in the case of a teen-community website. Therefore, internet platforms and websites with age restrictions can enforce these restrictions by applying range proofs on the age contained within an eID. The COP Initiative was launched in 2008 by the International Telecommunication Union (ITU) with the objective of protecting children in the online world.

Range proofs also have an important implication in e-cash and in e-auction scenarios, where participants are required to prove solvency, with the constraint that they are not willing to provide the exact amount present in their bank accounts for obvious privacy concerns. Lastly, some cryptographic protocols require the freshness of some confidential timestamps to be checked, which is a direct application of range proofs.

Regarding MRTDs, privacy is a big concern for their owners. For instance in Switzerland, 49.9% of electors voted against the introduction of the biometric passport on the 17^{th} May 2009 [FCoS09], presumably for privacy reasons. The importance of providing a good survey on the standards for MRTDs (such as the EAC) is motivated by the fact that the available standards neglect privacy and revocation issues. This carelessness induces serious security threats. In the case of terminal revocation, any stolen border patrol reader will be able to stealthy read all the private information contained in passports without the consent of the holder nor his awareness. Stolen data can help an attacker in the process of identity theft, and even allow him to circumvent biometric protections such as face and fingerprint recognition. Furthermore, a stolen terminal can be set to collect the data of all surrounding MRTDs, which will allow an attacker to build a rogue database of stolen identities. Lastly, a stolen terminal coupled with a monitoring system gives an attacker the ability to monitor a location for specific targets, whether specific individuals or some specific group of persons (selected, for instance, by nationality). This last threat is of major concern as it could be the scenario of an untraceable terrorist attack. Assume that an attacker chooses to target citizens of a specific country, and sets its system in a large transit hub (such as an airport) or in a large transport vehicle (such as an airplane). The rogue terminal will scan its surroundings, and when the requirements set by the attacker are met (for instance a large group of a targeted nationality), the terrorist attack would be triggered. Moreover, once the system of this scenario has been set up, the attacker can walk away freely before the attack is triggered. Hence, the terminal revocation problem raises the issue of targeted terrorist attacks.

1.2 Results and Contributions

This thesis incorporates the results of five publications from the author [CCs08, CLs10, CLZ12, CV09, Cha13] (see Section 1.4), together with some extensions, corrections, and unpublished results.

Regarding set membership proofs, this thesis presents several efficient interactive solutions, where "interactive" informally means that both participants exchange messages to complete the protocol. The first solution is a protocol based on the Boneh-Boyen signature scheme [BB04], which remains at the time of writing the most efficient protocol for set membership proofs. This thesis also shows that alternative signature schemes can be employed, by providing a set membership proof based on the Camenisch-Lysyanskaya signature scheme [CL02b]. Although the idea of using the alternative signature scheme from Camenisch-Lysyanskaya was presented in [CCs08], the construction and details of the protocol are exclusively presented in this thesis. Furthermore, this thesis also provides a general construction for any secure signature scheme. Lastly and as described in [CL02a], this thesis shows that set membership proofs can be based on cryptographic accumulators instead of signature schemes, with the example of the Camenisch-Lysyanskaya accumulator.

This thesis provides one of the first classifications of range proofs. Then, with the use of the set membership proofs mentioned above, interactive range proofs are constructed for ranges of the type $\left[0, u^{\ell}\right)$, where the elements of the range are decomposed in base u. This thesis provides a concrete construction with the Boneh-Boyen based set membership proof, as well as a general protocol for any set membership proof. This thesis then argues how to use known methods (OR and AND compositions) in order to handle arbitrary ranges. Moreover, this thesis exclusively provides the details for handling arbitrary ranges with the AND composition. The details for the OR composition have been set aside as several drawbacks were overlooked in [CCs08] and are explained in this thesis. Another unpublished result is the construction of a range proof based on a variant of the set membership proof of Arfaoui et al. [ALT+15a]. It should be noted that this last protocol can be deduced from our general protocol.

Furthermore, this thesis shows that number consecutiveness in range proofs can be further exploited with a better decomposition. This thesis improves the decomposition introduced in [LAN02] to obtain a sumset representation of integer intervals. It should also be noted that the bound for the decomposition achieved in this thesis is better than the published results from [CLs10]. Thus, by combining this new representation of integers with novel properties and theorems from the number theory field, as well as new results in additive combinatorics, this thesis provides the most efficient interactive range proof.

As some e-services forbid interactions, this thesis develops the first efficient and secure non-interactive range proof, without random oracles, which has been published in [CLZ12]. Note that "non-interactive" protocols informally restrict communications to one single transmitted message. Furthermore, "random oracles" are theoretical entities that map given inputs to

perfectly random outputs. Random oracles are often employed to prove the security of non-interactive protocols. However their use is controversial as some security issues regarding their concrete implementation were raised in [CGH98, CGH04]. Note as well that minor details from [CLZ12] are corrected in this thesis. This thesis also shows that one of the first attempts to produce a secure and efficient non-interactive range proof [YHM+09] is intrinsically insecure.

In the case of MRTDs, this thesis first recalls the threats linked to Radio Frequency IDentification (RFID) chips, which were chosen as a hardware component in the implementation of MRTDs. RFID chips, also called RFID tags, are hardware components that use electromagnetic fields to wirelessly transmit data. This thesis then surveys the two existing standards for MRTDs and their updates. The first standard [ICAO08] was produced by the International Civil Aviation Organization (ICAO) and this thesis shows that this standard possesses serious weaknesses regarding the data privacy of MRTDs. This thesis then surveys the alternative standard proposed by the German Federal Office for Information Security (BSI), called the Extended Access Control (EAC), at the request of the European Union. Despite some security advantages such as anti-cloning protection, this thesis shows that privacy concerns remain the same in the first version of the EAC [BSI08a], as in the ICAO standard. Finally, this thesis surveys the current EAC version (EACv2 [BSI15a, BSI15b, BSI15c, BSI15d]) and demonstrate that despite some claims [Nit09] as to the security improvements it brings, several flaws regarding privacy are still of concern, notably the terminal revocation problem. These surveys are updated to match the latest version of the EAC (February 2015), as the versions studied in the initial publications [CV09, Cha13] precede the latest version.

Lastly, this thesis provides several solutions as to how to enhance the different standards for MRTDs. The first solution that is provided, performs a minor hardware upgrade on MRTDs with the introduction of an RFID switch, that will allow the use of the RFID chip only when needed. The second solution that this thesis suggests in [CV09], was to replace some components of the ICAO standard with its more secure counterpart described in the EACv2. This suggestion was accepted in February 2013 by the ICAO working group in charge of the ICAO standard [ICAO13], and will be enforced in January 2018. This thesis also suggests a minor behavioral improvement for holders of MRTDs, to reduce the threat of terminals with expired certificates. As MRTDs only have an approximation of the current date, this minor behavioral improvement consists of updating this approximative date more often, especially before traveling. Finally, the major contribution that this thesis provides regarding MRTDs, is the elaboration of a significant solution for terminal revocation, that only require upgrading the underlying protocol. The details of this solution are provided as an extension of [Cha13] and its main idea consists of enforcing terminal collaboration in order to achieve the authentication of terminals.

1.3 Thesis Outline

In order to explain the core elements of this thesis, ***Chapter 2*** introduces the required essential ***preliminaries***. Notations and some basic definitions are provided in Section 2.1. The security models of the protocols described in this thesis, as well as their security assumptions, are explained and defined in Section 2.2. Section 2.3 is devoted to defining proofs, arguments, proofs of knowledge, and their corresponding security properties, which constitute set membership and range proofs. Section 2.4 describes all of the cryptographic primitives that are used as building blocks for our solutions. Section 2.5 is devoted to threshold cryptography, as it is the basis of our solution for terminal revocation.

The core of this thesis is then divided into two main parts. **Part I** focuses on set membership and range proofs. **Part II** studies the case of the Extended Access Control and Machine Readable Travel Documents (MRTDs).

Part I

Chapter 3 *provides results regarding* ***set membership proofs***.
The description and definition of this primitive is given in Section 3.1. Section 3.2 describes prior and related work. Then, the Boneh-Boyen signature scheme based set membership proof is explained in Section 3.3, together with its security proof and efficiency analysis. Set membership proofs based on alternative signature schemes are discussed in Section 3.4, with a concrete solution based on the Camenisch-Lysyanskaya signature scheme, and with a generalization for any signature scheme. Cryptographic accumulators based set membership proofs are explained in Section 3.5 with the example of the Camenisch-Lysyanskaya dynamic accumulator.

Chapter 4 *focuses on results for* ***interactive range proofs***.
The description and definition of this primitive is given in Section 4.1. Section 4.2 describes prior work, related work, and provides a classification for interactive range proofs. Section 4.3 is dedicated to our results regarding interactive range proofs based on set membership proofs, including a range proof for range $[0, u^{\ell})$ using our Boneh-Boyen signature scheme based set membership proof, its security proof and efficiency analysis, methods on how to handle arbitrary ranges with their respective security and efficiency analysis, a description of how to construct a range proof based on any secure set membership proof, and an example of such a construction based on a variant of the Arfaoui et al. set membership proof. Section 4.4 explains and proves the decomposition of integer intervals into a sumset representation. The sumset based range proof, together with its security and efficiency analysis, is thus described in Section 4.5, which concludes with a concrete example and efficiency comparisons between state of the art interactive range proofs.

Chapter 5 *explains the result of this thesis regarding* ***non-interactive range proofs****.*
The explanation of this primitive and its definitions are given in Section 5.1. Section 5.2 gives an overview of prior and more recent work, as well as related work. Furthermore, the Yuen et al. non-interactive range proof is proven insecure in Section 5.3. Section 5.4 provides an equality subargument (between a lifted BBS encryption and a knowledge commitment) and its security proof, which are necessary for our non-interactive range proof. The latter is presented in Section 5.5, together with its security and efficiency proofs.

Part II

Chapter 6 *surveys* ***Machine Readable Travel Documents*** *(MRTDs) and their related standards.*
After a brief introduction in Section 6.1, Section 6.2 presents prior and related work. The threats relating to the standard for RFID chips are recalled in Section 6.3. Section 6.4 surveys the ICAO standard for MRTDs. Section 6.5 briefly surveys the first version of the EAC standard for MRTDs. More details on the EAC standard are provided with a survey of its second version in Section 6.6. The conclusions of these surveys are provided in Section 6.7.

Chapter 7 *mainly aims at providing* ***enhancements to the EAC*** *standard.*
After a brief introduction in Section 7.1, Section 7.2 presents prior and related enhancements for MRTDs. Section 7.3 recommends the incorporation of a small hardware improvement in the form of an RFID switch in new MRTDs. Section 7.4 explains an enhancement of the ICAO standard, which has been accepted by the relevant ICAO working group. Section 7.5 provides a small recommendation on behavioral practices for MRTDs holders. Section 7.6 explains a suggested solution, together with its security and efficiency analysis, for the resolution of the terminal revocation problem.

Lastly, the ***conclusions*** of this thesis are provided in ***Chapter 8***.

1.4 List of Publications by the Author

The publications by the author that were included in this thesis, are listed in the following. Each of them is described briefly and the contributions made by the author specified. Special presentations of these publications are also mentioned.

[CCs08] Jan Camenisch, Rafik Chaabouni, and abhi shelat[1].
Efficient protocols for set membership and range proofs.
In Josef Pieprzyk, editor, *ASIACRYPT*, volume 5350 of *Lecture Notes in Computer Science*, pages 234-252. Springer, 2008

[CV09] Rafik Chaabouni and Serge Vaudenay.
The extended access control for machine readable travel documents.
In Arslan Brömme, Christoph Busch, and Detlef Hühnlein, editors, *BIOSIG*, volume 155 of *LNI*, pages 93-103. GI, 2009.

[CLs10] Rafik Chaabouni, Helger Lipmaa, and abhi shelat[1].
Additive combinatorics and discrete logarithm based range protocols.
In Ron Steinfeld and Philip Hawkes, editors, *ACISP*, volume 6168 of *Lecture Notes in Computer Science*, pages 336–351. Springer, 2010.

[CLZ12] Rafik Chaabouni, Helger Lipmaa, and Bingsheng Zhang.
A non-interactive range proof with constant communication.
In Angelos D. Keromytis, editor, *Financial Cryptography*, volume 7397 of *Lecture Notes in Computer Science*, pages 179–199. Springer, 2012.

[Cha13] Rafik Chaabouni.
Solving terminal revocation in EAC by augmenting terminal authentication.
In Arslan Brömme and Christoph Busch, editors, *BIOSIG*, volume 212 of *LNI*, pages 273–280. GI, 2013.

Publications [CCs08, CLs10, CLZ12] relate to *set membership and range proofs*.
Publications [CV09, Cha13] relate to *Machine Readable Travel Documents (MRTDs) and the Extended Access Control (EAC)*.

Publication [CCs08] introduced the concept of signature based set membership proofs with efficient secure protocols, and used them to construct efficient secure interactive range proofs. Inspired by Berry Schoenmakers [Sch01, Sch05], the author proposed the application of a u-ary decomposition for range proofs, where the digits would be proven with a set membership proof. This publication was presented at the Asiacrypt 2008 conference.
Publication [CLs10] improved the concept of decomposition for range proofs, with a multi-base decomposition called the sumset representation of integer intervals. This resulted in

[1] Note that abhi shelat requires his name to be written in lower case.

an efficiency gain by a factor of roughly 2, when handling arbitrary ranges. The sumset decomposition is a generalization of the binary sumset decomposition introduced in [LAN02]. The author provided corrections to the general sumset decomposition as the generalization was intricate. This publication was presented at the ACISP conference in 2010.
Publication [CLZ12] proved the first attempt to produce a secure efficient non-interactive range proof without random oracles [YHM+09] to be insecure, and provided a novel protocol as an answer. The insecurity proof of [YHM+09] was achieved by the author. This publication was presented at the Financial Cryptography conference in 2012.

Publication [CV09] is a general survey of the standards for MRTDs. This publication also stated the remaining problems of these standards and suggested some improvements. The author studied the different standards, analyzed them, identified some of the remaining problems, and elaborated the suggested improvements. This publication was first presented at the BIOSIG conference in 2009. The author was then also invited by the Arab ICT Organization to present this publication at the *First Arab Forum* on "e-transactions Security & the Public Key Infrastructure (PKI)", in 2010.
Publication [Cha13] provided a solution to the terminal revocation problem present in the EAC. This solution is based on the threshold RSA signature of Shoup [Sho00] and only requires a software upgrade. This single-authored publication was presented at the BIOSIG conference in 2013.

Chapter 2

Preliminaries

This chapter will introduce the various notations and definitions, as well as the important preliminaries necessary for understanding the construction of our protocols. Section 2.1 introduces general notations as well as the Bachmann-Landau notations (also known as the big O notation), defines the notion of indistinguishability, recalls bilinear groups, and lastly defines some specific notions of combinatorics. The security models, which prove the security of the solutions presented in this thesis, are explained in Section 2.2. The definitions relating to cryptographic proofs, arguments, and proofs of knowledge are provided in Section 2.3, together with some definitions for related security properties (zero-knowledge, witness hiding, witness indistinguishability). Section 2.4 recalls some common definitions regarding different protocols and building blocks used in cryptography, as well as some specific protocols used in this thesis as building blocks. Lastly, Section 2.5 is dedicated to the notion of threshold cryptography, which will be essential for the solution provided in Chapter 7.

Beside the original citations, three main references were used to compile the majority of the definitions and notions presented in this chapter:

- *"A Classical Introduction to Cryptography"* by Serge Vaudenay [Vau06],
- the *"Encyclopedia of Cryptography and Security"* [vTJ11],
- and the *"Foundations of Cryptography"* by Oded Goldreich [Gol01, Gol04].

2.1 Notations and Definitions

The goal of this section is to clarify notations and to recall some basic definitions in mathematics and in cryptography, that will be necessary for understanding the following chapters.

Regarding notations, probabilistic polynomial-time algorithms will be denoted by *PPT* algorithms. Interactive Turing Machines will be denoted as *ITM* (see Section 2.3.1). The size of

a set $\mathbb{F}$ will be denoted $|\mathbb{F}|$. The absolute value of an element x will be denoted $|x|$. The bit length of an element x will be denoted $\|x\|$. The bit length of an element taken in a group $\mathbb{G}$ will be denoted $\|\mathbb{G}\|$. The primordial security parameter will be denoted by κ. In some cases, a secondary security parameter will be needed. It will be denoted k and will usually be dependent on κ. Moreover, security parameters will be expressed in the *unary numeral* system, where a number n is represented by a n-long sequence of 1, denoted by 1^n. $\mathbb{Z}^+$ denotes the set of positive integers excluding 0. For $n \in \mathbb{N}$, $\mathbb{Z}_n$ denotes the cyclic group of order n. $\mathbb{Z}_n^*$ denotes the multiplicative group of all invertible elements in $\mathbb{Z}_n$. The set $\{x_i\}_{i\in\mathbb{I}}$ is the set of elements x_i indexed by the countable index set $\mathbb{I}$.

The next two definitions recall the notions of *safe prime* and of *quadratic residuosity*. It is important to not confuse *safe primes* with *strong primes*. A strong prime p is a large prime such that $p-1$ and $p+1$ have a large prime factor. Moreover, the large prime factor q of $p-1$ is such that $q-1$ also has a large prime factor.

Definition 2.1 (Safe prime)
A safe prime *p is a prime number equal to $p = 2q+1$, where q is another prime number. In the case of a safe prime $p = 2q+1$, q is called a* Sophie-Germain prime.

Definition 2.2 (Quadratic residuosity)
A quadratic residue *modulo n is an element $a \in \mathbb{Z}_n^*$ such that $\exists b \in \mathbb{Z}_n^* : a \equiv b^2 \pmod{n}$. The set of all quadratic residues modulo n is denoted $QR_n \subseteq \mathbb{Z}_n^*$.*

2.1.1 Bachmann-Landau notations

The *Bachmann-Landau notations* provide notations that describe the asymptotic growth of functions. Five distinctive asymptotical growths for a function $f(n)$ can be distinguished, regarding the function $g(n)$ to which it is being compared to: $O\big(g(n)\big)$, $o\big(g(n)\big)$, $\Omega\big(g(n)\big)$, $\omega\big(g(n)\big)$, and $\Theta\big(g(n)\big)$.

Definition 2.3 ($O(\cdot)$ notation)
Let $f(n)$ and $g(n)$ be two functions. The notation $f(n) = O\big(g(n)\big)$ or $f(n) \in O\big(g(n)\big)$ means that f is asymptotically dominated by g, up to a constant factor:

$$\boxed{\exists A > 0,}\ \exists n_0,\ \forall n > n_0 \ :\ \big|f(n)\big| \leqslant A \cdot \big|g(n)\big|.$$

Definition 2.4 ($o(\cdot)$ notation)
Let $f(n)$ and $g(n)$ be two functions. The notation $f(n) = o\big(g(n)\big)$ or $f(n) \in o\big(g(n)\big)$ means that f is asymptotically dominated by g:

$$\boxed{\forall A > 0,}\ \exists n_0,\ \forall n > n_0 \ :\ \big|f(n)\big| \leqslant A \cdot \big|g(n)\big|.$$

Note that $o\big(g(n)\big)$ implies $O\big(g(n)\big)$.

Definition 2.5 ($\Omega(\cdot)$ notation)
Let $f(n)$ and $g(n)$ be two functions. The notation $f(n) = \Omega\big(g(n)\big)$ or $f(n) \in \Omega\big(g(n)\big)$ means that f asymptotically dominates g, up to a constant factor:

$$\boxed{\exists A > 0,}\ \exists n_0,\ \forall n > n_0 \ :\ \big|f(n)\big| \geqslant A \cdot \big|g(n)\big|.$$

Note that $f(n) = \Omega\big(g(n)\big) \Longleftrightarrow g(n) = O\big(f(n)\big)$.

Definition 2.6 ($\omega(\cdot)$ notation)
Let $f(n)$ and $g(n)$ be two functions. The notation $f(n) = \omega\big(g(n)\big)$ or $f(n) \in \omega\big(g(n)\big)$ means that f asymptotically dominates g:

$$\boxed{\forall A > 0,}\ \exists n_0,\ \forall n > n_0 \ :\ \big|f(n)\big| \geqslant A \cdot \big|g(n)\big|.$$

Note that $\omega\big(g(n)\big)$ implies $\Omega\big(g(n)\big)$.

Definition 2.7 ($\Theta(\cdot)$ notation)
Let $f(n)$ and $g(n)$ be two functions. The notation $f(n) = \Theta\big(g(n)\big)$ or $f(n) \in \Theta\big(g(n)\big)$ means that f is asymptotically bounded by g:

$$\boxed{\exists A_1 > 0,\ \exists A_2 > 0,}\ \exists n_0,\ \forall n > n_0 \ :\ A_1 \cdot g(n) \leqslant f(n) \leqslant A_2 \cdot g(n).$$

Note that $\Theta\big(g(n)\big)$ implies both $O\big(g(n)\big)$ and $\Omega\big(g(n)\big)$.

2.1.2 Indistinguishability

The *indistinguishability* property is an essential measure that is used for security proofs. Informally, this measure aims to assess the distance between two distributions. In order to do so, it relies on a distinguisher algorithm and on the notion of negligible functions.

Definition 2.8 (Family)
A family *is a sequence of random variables indexed by a countable index set. For instance the family $X = \{X_i\}_{i \in \mathbb{I}}$ is the sequence of random variables X_i indexed by the countable index set $\mathbb{I}$.*

Definition 2.9 (Distinguisher)
A distinguisher *$\mathscr{D}$ is a PPT algorithm that takes some input and outputs either 1 or 0. It is used to compare two given families X, Y of random variables with identical index set I. The distance between X and Y is the family of $\Pr[\mathscr{D}(X_n, 1^n) = 1] - \Pr[\mathscr{D}(Y_n, 1^n) = 1]$ for $n \in I$. When Y is the family of random variables with uniform distribution, this becomes a measure of the randomness of X. Furthermore, the notation $\mathscr{D}(X_n, 1^n)$ will be used to include an emphasis on the security parameter and on the index set.*

Definition 2.10 (Negligible)

A function $f : \mathbb{N} \to [0,1]$ is said to be negligible in n if for every positive polynomial $p(\cdot)$, there exists a $n_0 \in \mathbb{N}$ such that for every $n > n_0$, the following holds:

$$f(n) < \frac{1}{p(n)}.$$

Indistinguishability between two families reflects how close they are in terms of their respective random variable. Three levels of indistinguishability precision can be differentiated: computational, statistical and perfect.

Definition 2.11 (Computational Indistinguishability)

Two families X_n, Y_n with identical index set $\mathbb{N}$ are said to be computationally indistinguishable *if for any distinguisher $\mathscr{D}$, the function $|\Pr[\mathscr{D}(X_n, 1^n) = 1] - \Pr[\mathscr{D}(Y_n, 1^n) = 1]|$ is negligible in n. In other words, for every positive polynomial $p(\cdot)$, there exists an index $n_0 \in \mathbb{N}$ such that for every $n > n_0$, the following holds:*

$$\left|\Pr\left[\mathscr{D}\left(X_n, 1^n\right) = 1\right] - \Pr\left[\mathscr{D}\left(Y_n, 1^n\right) = 1\right]\right| < \frac{1}{p(n)}.$$

To express that two families are computationally indistinguishable, the following notation is used:

$$\{X_n\} =_c \{Y_n\}.$$

Definition 2.12 (Statistical Indistinguishability)

Let X_n, Y_n be two families over a finite domain $\mathscr{X}$ and with identical index set $\mathbb{N}$. These families are said to be statistically indistinguishable *if their statistical distance is negligible in n. Hence, the two families X_n, Y_n are statistically indistinguishable if, for every positive polynomial $p(\cdot)$, there exists an index $n_0 \in \mathbb{N}$ such that for every $n > n_0$ and every element $a \in \mathscr{X}$, the following holds:*

$$\frac{1}{2} \sum_{a \in \mathscr{X}} |\Pr[X_n = a] - \Pr[Y_n = a]| < \frac{1}{p(n)}.$$

To express that two families are statistically indistinguishable, the following notation is used:

$$\{X_n\} =_s \{Y_n\}.$$

Definition 2.13 (Perfect Indistinguishability)

Two families X_n, Y_n with identical index set $\mathbb{N}$ are said to be perfectly indistinguishable *if for every element $a \in X_n$ and every index $n \in \mathbb{N}$, the following holds:*

$$\Pr[X_n = a] = \Pr[Y_n = a].$$

To express that two families are perfectly indistinguishable, the following notation is used:

$$\{X_n\} =_p \{Y_n\}.$$

2.1.3 Bilinear Groups

Bilinear groups are special algebraic groups that define a function e, called a bilinear map or a pairing function. The properties of this latter function are essential in many cryptographic applications. Let $\mathsf{PG}_a(1^\kappa)$ be an asymmetric bilinear group generator that on input 1^κ outputs a description of a bilinear group $\mathsf{param}_{\mathsf{bp}} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathrm{T}, e) \leftarrow \mathsf{PG}_a(1^\kappa)$ such that p is a κ-bit prime, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_\mathrm{T}$ are multiplicative cyclic groups of prime order p, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_\mathrm{T}$ is an admissible bilinear map (pairing). Let $\mathbb{G}_1^* = \mathbb{G}_1 \setminus \{1\}$, $\mathbb{G}_2^* = \mathbb{G}_2 \setminus \{1\}$ and let $g_1 \in \mathbb{G}_1^*$, $g_2 \in \mathbb{G}_2^*$ be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. The admissible bilinear map e is such that

- for all $a, b \in \mathbb{Z}_p$ it holds that $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$;
- $e(g_1, g_2) \neq 1$ generates $\mathbb{G}_\mathrm{T}$;
- it is efficient to decide membership in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_\mathrm{T}$;
- group operations and the bilinear map are efficiently computable;
- generators are efficiently samplable, and the descriptions of the groups and group elements are $O(\kappa)$ bit long.

Furthermore, in the case of symmetric bilinear groups, $\mathbb{G}_2 = \mathbb{G}_1$. Moreover, the (symmetric) bilinear group generator will be denoted $\mathsf{PG}(1^\kappa)$. If not specified, bilinear groups will imply the symmetric case. It is important to notice that following the results of [GPS08, AMOR14, GKZ14], symmetric bilinear groups are now considered either insecure or unpractical, and therefore the asymmetric case should be enforced in practice. This thesis kept the use of symmetric pairings for Chapter 3 and Chapter 4 in an effort to remain easy to understand. Their use was nevertheless dropped in Chapter 5.

2.1.4 Combinatorics

Chapter 5 requires an understanding of several notions of combinatorics and additive combinatorics that are explained hereafter. The following additive combinatorics definitions are taken from [TV06]. Let Λ, Λ_1 and Λ_2 be subsets of some additive group, such as $\mathbb{Z}$ or $\mathbb{Z}_n$.

Definition 2.14 (Sum set)
The sum set *of* Λ_1 *and* Λ_2 *is* $\Lambda_1 + \Lambda_2 = \{\lambda_1 + \lambda_2 : \lambda_1 \in \Lambda_1 \wedge \lambda_2 \in \Lambda_2\}$.

Definition 2.15 (Difference set)
The difference set *of* Λ_1 *and* Λ_2 *is* $\Lambda_1 - \Lambda_2 = \{\lambda_1 - \lambda_2 : \lambda_1 \in \Lambda_1 \wedge \lambda_2 \in \Lambda_2\}$.

Definition 2.16 (Iterated sumset)
The iterated sumset *of* Λ *by* $s \in \mathbb{Z}^+$ *is* $s\Lambda = \left\{\sum_{i=1}^{s} \lambda_i : \lambda_i \in \Lambda\right\}$.

Definition 2.17 (Dilation)
The dilation *of* Λ *by* $s \in \mathbb{Z}^+$ *is* $s \cdot \Lambda = \{s\lambda : \lambda \in \Lambda\}$.

Definition 2.18 (Restricted sumset)
The restricted sumset *of* Λ *is* $2\hat{}\Lambda = \{\lambda_1 + \lambda_2 : \lambda_1 \in \Lambda \wedge \lambda_2 \in \Lambda \wedge \lambda_1 \neq \lambda_2\} \subseteq \Lambda + \Lambda$.

Definition 2.19 (Progression-free set)
A set $\Lambda = \{\lambda_i\} \subset \mathbb{Z}^+$ *is a* progression free set, *if no three of its elements are in arithmetic progression, so that* $\lambda_i + \lambda_j = 2\lambda_k$ *only if* $i = j = k$.

Let $r_3(N)$ denote the cardinality of the largest progression free set that belongs to $\{1,\ldots,N\}$. Elkin showed in [Elk10] that

$$r_3(N) = \Omega\left(\left(N \cdot \log_2^{1/4} N\right)/2^{2\sqrt{2\log_2 N}}\right).$$

It is also known from [San11] that $r_3(N) = O(N(\log\log N)^5/\log N)$. Thus, the minimal N such that $r_3(N) = n$ is $\omega(n)$, while according to Elkin, $N = n^{1+o(1)}$.

Theorem 2.1 ([Lip12a](Theorem 1, Section 3))
For any $n > 0$, *there exists* $N = n^{1+o(1)}$, *such that* $\{1,\ldots,N\}$ *contains a progression free subset* Λ *of odd integers of cardinality* n.

2.2 Security Models

Cryptographic primitives and protocols should convince users that they are secure. In order to confirm that, exact mathematical proofs are desirable. When achieved, they are called security proofs. Unfortunately, perfect security is not always feasible [Gol01] and often leads to inefficient protocols. To produce efficient but nonetheless secure cryptographic protocols, the security proofs rely on security models that define time and computational restrictions, as well as any additional advantages. Hereafter, three of the most used models are explained. The *standard model* is the most common model used. It focuses on time and computational restrictions by means of complexity assumptions called *computational hardness assumptions*. The *random oracle model* introduces the advantage of random functions. Note that this is different from random sampling. The *common reference string model* provides a specific bit string to participants of a cryptographic protocol. Finally, *knowledge assumptions* are non-standard restrictions that give evidence about the "knowledge" of the prover.

2.2.1 Standard Model

The *standard model*, as defined by Naccache in the "Encyclopedia of Cryptography and Security" [vTJ11], assumes that the adversaries are only probabilistic and computationally bounded

in polynomial time. They are thus called probabilistic polynomial-time (PPT) adversaries. Therefore, this model defines security based on mathematical problems which are considered hard to solve in polynomial time. To meet this criteria, these problems rely on complexity assumptions called *computational hardness assumptions*. The security is then guaranteed as long as its corresponding computational hardness assumptions hold (i.e. as long as the corresponding mathematical problems remain hard to solve). Security is thus defined with arguments of security instead of proofs of security. Nevertheless, the cryptographic literature refers to them as security proofs. This abuse of terminology will be kept in this thesis as well.

2.2.2 Computational Hardness Assumptions

Computational hardness assumptions, as mentioned above, are complexity assumptions where some specific mathematical problems are assumed to be computationally hard to resolve in polynomial time. Alternative complexity assumptions are sometimes used such as *decisional hardness assumptions*, where the veracity of a statement is computationally hard to attest in polynomial time.

Computational Hardness Assumption A *computationally hard problem* is a mathematical problem that is computationally hard to solve under certain parameters, generated by the algorithm of the challenger C_{gen}. The *computational hardness assumption* is thus defined regarding a computationally hard problem and C_{gen}. A computational hardness assumption states that given C_{gen} and for any PPT adversary $\mathscr{A}$, the adversary is able to solve the corresponding *computationally hard problem* with a probability that is negligible in terms of the security parameter κ.

Decisional Hardness Assumption A *decisional hard problem* is a mathematical problem that requires a distinguisher $\mathscr{D}$ to decide on the veracity of a statement under certain parameters, generated by the algorithm of the challenger C_{gen}. The *decisional hardness assumption* is defined regarding a decisional hard problem, C_{gen}, and a distinguisher $\mathscr{D}$. A decisional hardness assumption states that given C_{gen} and for any distinguisher $\mathscr{D}$, the distinguisher is able to solve the corresponding *decisional hard problem* with a distance probability that is negligible in terms of the security parameter κ.

Historically, the two best studied computationally hard problems in cryptography are the factorization problem and the discrete logarithm problem (DLog problem). Both of these problems led to the establishment of subsequent computationally hard problems.

Computationally Hard Problem 2.1 (Factorization Problem)
Given n, the product of two prime numbers, find its factorization as $n = p \cdot q$. The related computational hardness assumption is thus called the factorization assumption.

Computationally Hardness Assumption 2.1 (Factorization Assumption)
Let $\mathrm{C}_{\mathrm{gen}}$ *output* (n,p,q) *on input* 1^κ *such that* $\|p\| = \|q\| = \kappa$ *and* $n = pq$. *The* factorization assumption *relative to* $\mathrm{C}_{\mathrm{gen}}$ *states that for any PPT adversary* $\mathscr{A}$,

$$Pr\Big[(n,p,q) \leftarrow \mathrm{C}_{\mathrm{gen}}(1^\kappa),\ (p,q) \leftarrow \mathscr{A}(n) :\ n = pq\Big] \text{ is negligible in } \kappa.$$

Note that the 768-bit number named "RSA-768" (from the RSA cryptosystem) was factored in 2009 by Kleinjung et al. in [KAF+10]. Thus, 1024-bit numbers are now considered at risk. Furthermore, the now famous RSA public key cryptosystem [RSA78] (see Section 2.4.3 for a reminder of public key cryptosystems) was introduced in 1978 by Rivest, Shamir and Adleman. The computational hardness assumption associated with this cryptosystem, was followed by several important ones including the *strong RSA assumption*, which is necessary for the set membership proofs presented in sections 3.4 and 3.5. Recall that an RSA modulus n is equal to the product of two large primes p and q, hence $n = pq$. The RSA ciphertext c of a message m is obtained by computing $c = m^e \pmod{n}$, where e is coprime to $\big((p-1)(q-1)\big)$. Deciphering c is achieved by computing $m = c^d \pmod{n}$, where d is part of the secret key as $d = e^{-1} \pmod{(p-1)(q-1)}$.

Computationally Hard Problem 2.2 (RSA Problem [RSA78])
Given an RSA public key (n,e) *with* $e > 1$, *and an RSA ciphertext* c, *find a message* m *such that* $c = m^e \pmod{n}$.

Computationally Hardness Assumption 2.2 (RSA Assumption)
Let $\mathrm{C}_{\mathrm{gen}}$ *output* (n,p,q,e,c) *on input* 1^κ *such that* p,q *are primes,* $\|p\| = \|q\| = \kappa$, $n = pq$, $e > 1$ *is coprime to* $\big((p-1)(q-1)\big)$, *and* $c \in_R \mathbb{Z}_n$. *The* RSA assumption *relative to* $\mathrm{C}_{\mathrm{gen}}$ *states that for any PPT adversary* $\mathscr{A}$,

$$Pr\Big[(n,p,q,e,c) \leftarrow \mathrm{C}_{\mathrm{gen}}(1^\kappa),\ (m) \leftarrow \mathscr{A}(n,e,c) :\ c = m^e \pmod{n}\Big] \text{ is negligible in } \kappa.$$

Computationally Hard Problem 2.3 (Strong RSA Problem [FO97])
Given an RSA modulus n *and an RSA ciphertext* c, *find a pair* $(m,\ e)$ *such that* $e > 1$ *and* $c = m^e \pmod{n}$. *The related computational hardness assumption is called the* Strong RSA assumption.

Computationally Hardness Assumption 2.3 (Strong RSA Assumption)
Let $\mathrm{C}_{\mathrm{gen}}$ *output* (n,p,q,c) *on input* 1^κ *such that* p,q *are primes,* $\|p\| = \|q\| = \kappa$, $n = pq$, *and* $c \in_R \mathbb{Z}_n$. *The* strong RSA assumption *relative to* $\mathrm{C}_{\mathrm{gen}}$ *states that for any PPT adversary* $\mathscr{A}$,

$$Pr\Big[(n,p,q,c) \leftarrow \mathrm{C}_{\mathrm{gen}}(1^\kappa),\ (m,e) \leftarrow \mathscr{A}(n,c) :\ c = m^e \pmod{n} \ \wedge\ e > 1\Big] \text{ is negligible in } \kappa.$$

Another subsequent result is the *decisional composite residuosity assumption*, which was introduced by Paillier in [Pai99] as a conjecture.

Decisional Hard Problem 2.4 (DCR Problem [Pai99])
Assuming that n is an RSA modulus ($n = pq$), and given a number x, decide if x is an n^{th} residue modulo n^2, that is if there exists a number $y \in \mathbb{Z}^*_{n^2}$ such that $x = y^n \pmod{n^2}$. The related decisional hardness assumption is called the DCR assumption.

Decisional Hardness Assumption 2.4 (DCR Assumption [Pai99])
Let $\mathrm{C}_{\mathrm{gen}}$ output (n, p, q, y, z) on input 1^κ such that p, q are primes, $\|p\| = \|q\| = \kappa$, $n = pq$, and y, z are uniformly random variables taken independently in $\mathbb{Z}_{n^2}$. Let $x = y^n \pmod{n^2}$. The DCR assumption *relative to $\mathrm{C}_{\mathrm{gen}}$, states that the distributions x and z are computationally indistinguishable in terms of the security parameter κ.*

The discrete logarithm (DLog) problem is defined as follows:

Computationally Hard Problem 2.5 (Discrete Logarithm (DLog) Problem)
Given a finite (multiplicative) abelian group $\mathbb{G}$, a generator g of $\mathbb{G}$, and a random element $y \in \mathbb{G}$, find the smallest integer x such that $y = g^x$. Moreover, x is called the discrete logarithm of y as $x = \log_g y$. The related computational hardness assumption is called the DLog assumption.

Computationally Hardness Assumption 2.5 (DLog Assumption)
Let $\mathrm{C}_{\mathrm{gen}}$ output $(\mathbb{G}, p, g, y)$ on input 1^κ such that $\|p\| = \kappa$, g has order p, g generates $\mathbb{G}$, and $y \in_R \mathbb{G}$. The DLog assumption *relative to $\mathrm{C}_{\mathrm{gen}}$ states that for any PPT adversary $\mathcal{A}$,*

$$Pr\left[(\mathbb{G}, p, g, y) \leftarrow \mathrm{C}_{\mathrm{gen}}(1^\kappa),\ x \leftarrow \mathcal{A}(\mathbb{G}, p, g, y) :\ y = g^x\right] \text{ is negligible in } \kappa.$$

Although Shoup showed in [Sho97] that the computational complexity in recovering x is of $\Omega\left(\sqrt{q}\right)$ where $q$is the largest prime dividing the group order when only using group operations, Bouvier et al. announced in [BGI+14], that they were able to solve the DLog problem for a group $\mathbb{Z}_p$, with p being a 596-bit safe prime by using the specific properties of $\mathbb{Z}_p$ (and by using the number field sieve algorithm).

Diffie and Hellman introduced in [DH76] the Diffie-Hellman (DH) problem. Initially, this problem related to their key agreement protocol, where two parties interact with each other in order to set up a common secret key over some insecure communication channel. Moreover, the DH problem opened the way to several related problems that were used in numerous protocols, including the set membership proof presented in Section 3.3, and the range proofs presented in Chapter 4 and in Chapter 5.

Computationally Hard Problem 2.6 (Diffie-Hellman (DH) Problem)
Given a finite (multiplicative) abelian group $\mathbb{G}$, a generator g of $\mathbb{G}$, and random elements $g^x,\ g^y \in \mathbb{G}$, compute g^{xy}. The related computational hardness assumption is called the DH assumption.

Computationally Hardness Assumption 2.6 (DH Assumption)
Let $\mathrm{C}_{\mathrm{gen}}$ output $(\mathbb{G}, p, g, x, y)$ on input 1^κ such that $\|p\| = \kappa$, g has order p, g generates $\mathbb{G}$, and x, y are random elements in the group order. The DH assumption *relative to $\mathrm{C}_{\mathrm{gen}}$ states*

that for any PPT adversary $\mathscr{A}$*,*

$$Pr\left[(\mathbb{G},p,g,x,y)\leftarrow \mathrm{C}_{\mathrm{gen}}(1^{\kappa}),\ h\leftarrow \mathscr{A}(\mathbb{G},p,g,g^{x},g^{y}):\ h=g^{xy}\right]\ \textit{is negligible in}\ \kappa.$$

The DH problem is considered hard for groups $\mathbb{G}=\mathbb{Z}_p^*$ with large prime p ([Vau06]), where the size of p should be as large as for the DLog problem.

Decisional Hard Problem 2.7 (Decisional Diffie-Hellman (DDH) Problem)
Given a finite (multiplicative) abelian group $\mathbb{G}$*, a generator* g *of* $\mathbb{G}$*, and three elements* $g^x,\ g^y,\ g^z\in\mathbb{G}$*, decide if* $z\stackrel{?}{=}xy$*. The related computational hardness assumption is called the* DDH assumption.

Decisional Hardness Assumption 2.7 (DDH Assumption)
Let $\mathrm{C}_{\mathrm{gen}}$ *output* $(\mathbb{G},p,g,x,y,z)$ *on input* 1^{κ} *such that* $\|p\|=\kappa$*,* g *has order* p*,* g *generates* $\mathbb{G}$*, and* x,y,z *are uniformly random variables taken independently in* $\mathbb{Z}_p$*. The* DDH assumption *relative to* $\mathrm{C}_{\mathrm{gen}}$ *states that the distributions* $\left(g^x,g^y,g^{xy}\right)$ *and* $\left(g^x,g^y,g^z\right)$ *are computationally indistinguishable in terms of the security parameter* κ*.*

It is important to note here, that the DDH assumption does not hold for $\mathbb{G}=\mathbb{Z}_p^*$, where p is prime, as the Legendre symbol, which identifies quadratic residues, provides information on the decisional problem $z\stackrel{?}{=}xy$. Nevertheless, if $\mathbb{G}$ is selected as the set of k^{th} power residues modulo a large prime p, such that $\frac{(p-1)}{k}$ is also a large prime, then the DDH assumption is believed to hold. These groups are called *Schnorr groups*. Furthermore, Schnorr groups with $k=2$ are the set of quadratic residues QR_p for a safe prime p.

Computationally Hard Problem 2.8 (q-Strong DH (q-SDH) Problem [BB04])
Assume that $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *are two cyclic groups of prime order* p*, where* $\mathbb{G}_1=\mathbb{G}_2$ *is possible. Assume that the two generators* g_1 *and* g_2 *respectively generate* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*. The* q-Strong Diffie-Hellman problem *consists of outputting a pair* $\left(c,g_1^{1/(x+c)}\right)$*, where* $c\in\mathbb{Z}_p^*$*, given the tuple* $\left(g_1,\ g_2,\ g_2^x,\ \ldots,\ g_2^{(x^q)}\right)$*, where* $x\in_R\mathbb{Z}_p^*$*. In order to define* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, the* q*-SDH problem is often associated with the asymmetric pairing generator* $\mathsf{PG}_a(1^{\kappa})$*. In the case of* $\mathbb{G}_1=\mathbb{G}_2$*, the* q*-SDH problem is thus associated with the symmetric pairing generator* $\mathsf{PG}(1^{\kappa})$*. The related computational hardness assumption is called the* q-SDH assumption.

Computationally Hardness Assumption 2.8 (q-SDH Assumption)
Let $\mathrm{C}_{\mathrm{gen}}$ *output* $(\mathbb{G}_1,\mathbb{G}_2,p,g_1,g_2,x,q)$ *on input* 1^{κ} *such that* p *is prime,* $\|p\|=\kappa$*,* g_1,g_2 *have order* p*,* g_1 *generates* $\mathbb{G}_1$*,* g_2 *generates* $\mathbb{G}_2$*,* $x\in_R\mathbb{Z}_p^*$*, and* $q\in\mathbb{N}$*. The* q-SDH assumption *relative to* $\mathrm{C}_{\mathrm{gen}}$ *states that for any PPT adversary* $\mathscr{A}$*,*

$$Pr\left[\begin{array}{c}(\mathbb{G}_1,\mathbb{G}_2,p,g_1,g_2,x,q)\leftarrow \mathrm{C}_{\mathrm{gen}}(1^{\kappa}),\ (c,s)\leftarrow \mathscr{A}\left(\mathbb{G}_1,\mathbb{G}_2,p,g_1,g_2,g_2^x,\ \ldots,\ g_2^{(x^q)}\right):\\ s=g_1^{1/(x+c)}\end{array}\right]$$

is negligible in κ*.*

A recent study by Cheon [Che06] shows a “weakness" in the q-SDH assumption. His results imply that the SDH problem has a computational complexity reduced by a factor of $\sqrt{q}$ from that of the discrete logarithm problem. Hence the generic computational complexity to recover x is $O\left(\sqrt{|\mathbb{G}_2|/q}\right)$ group operations instead of $\Omega\left(\sqrt{|\mathbb{G}_2|}\right)$ as claimed in [Sho97]. Nevertheless, this “weakness" is not relevant when q is a very small number compared to $|\mathbb{G}_2|$. In this thesis, it is assumed that q is less than 15 bits, whereas $|\mathbb{G}_2|$ is either greater than 256 bits (in the case of symmetric bilinear groups), or greater than 512 bits (in the asymmetric case).

Computationally Hard Problem 2.9 (Reverse Double Pairing Problem [AFG+10])
Assume that $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *are asymmetric bilinear groups. Given* $u, v \in \mathbb{G}_2$, *the* reverse double pairing problem *consists of finding elements* $s, t \in \mathbb{G}_1 \setminus \{1\}$ *such that* $e(s,u) = e(t,v)$. *The related computational hardness assumption is called the* reverse double pairing assumption.

Computationally Hardness Assumption 2.9 (Reverse Double Pairing Assumption [AFG+10])
Let $\mathrm{C}_{\mathrm{gen}}$ *output* $(\mathsf{param}_{\mathsf{bp}}, u, v)$ *on input* 1^κ *such that* $\mathsf{param}_{\mathsf{bp}} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathrm{T}, e) \leftarrow \mathsf{PG}_a(1^\kappa)$, *and* $u, v \in_R \mathbb{G}_2$. *The* reverse double pairing assumption *relative to* $\mathrm{C}_{\mathrm{gen}}$ *states that for any PPT adversary* $\mathscr{A}$,

$$Pr\left[\begin{array}{c}(\mathsf{param}_{\mathsf{bp}}, u, v) \leftarrow \mathrm{C}_{\mathrm{gen}}(1^\kappa),\ (s,t) \leftarrow \mathscr{A}(\mathsf{param}_{\mathsf{bp}}, u, v) : \\ e(s,u) = e(t,v)\ \wedge\ s,t \in \mathbb{G}_1 \setminus \{1\}\end{array}\right] \text{ is negligible in } \kappa.$$

Furthermore, Abe et al. showed in [AFG+10] that the DDH assumption implies the reverse double pairing assumption. Note that in [Gro11], Groth has referred to this assumption as the “computational double pairing assumption".

Decisional Hard Problem 2.10 (Decision Linear (DLIN) Problem [BBS04])
Given a finite (multiplicative) abelian group $\mathbb{G}$ *of prime order* p, *three generators* $u,\ v,\ h$ *of* $\mathbb{G}$, *and three group elements* $u^a,\ v^b,\ h^c \in G$, *decide if* $a + b \stackrel{?}{=} c \pmod{p}$. *The related decisional hardness assumption is called the* DLIN assumption.

Decisional Hardness Assumption 2.10 (DLIN Assumption)
Let $\mathrm{C}_{\mathrm{gen}}$ *output* $(\mathbb{G}, p, u, v, h, a, b, c)$ *on input* 1^κ *such that* p *is prime,* $\|p\| = \kappa$, u, v, h *are three random generators of* $\mathbb{G}$, *and* a, b, c *are uniformly random variables taken independently in* $\mathbb{Z}_p$. *The* DLIN assumption *relative to* $\mathrm{C}_{\mathrm{gen}}$ *states that the distributions* $\left(u^a, v^b, h^{a+b}\right)$ *and* $\left(u^a, v^b, h^c\right)$ *are computationally indistinguishable in* κ.

Here again, the DDH assumption implies the DLIN assumption, by solving the following request in the DLIN problem:

$$\left(u,\ v,\ h,\ u^a,\ v^b,\ h^c\right) \leftarrow \left(g^x,\ v \neq 1,\ g,\ g^z,\ 1,\ g^y\right).$$

The Λ-Power Symmetric Discrete Logarithm (Λ-PSDL) assumption is necessary for the protocols in Chapter 5. It was introduced by Lipmaa in [Lip12a] and originates from a related

assumption in [GJM02]. Assume that an asymmetric pairing generator $\mathsf{PG}_a(1^\kappa)$ outputs the parameters $\mathsf{param_{bp}} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathsf{PG}_a(1^\kappa)$. Assume that g_1 and g_2 are respectively generators of $\mathbb{G}_1$ and $\mathbb{G}_2$. Assume that $\Lambda = \{\lambda_i\}_{i\in\mathbb{Z}_n} \subset \mathbb{N}$ such that $\forall i < j$: $0 < \lambda_i < \lambda_j$, and that $x \in_R \mathbb{Z}_p^* \setminus \{1\}$.

Computationally Hard Problem 2.11 (Λ-PSDL Problem [Lip12a])
Given $\mathsf{param_{bp}}$ *and* $\left(g_1^{x^i},\ g_2^{x^i}\right)_{i\in\{0\}\cup\Lambda}$, *compute* x. *The related computational hardness assumption, called the* Λ-PSDL assumption, *states that* PG_a *is* Λ*-PSDL secure.*

Computationally Hardness Assumption 2.11 (Λ-PSDL Assumption [Lip12a])
Let PG_a *output* $(\mathsf{param_{bp}})$ *and* $\mathrm{C_{gen}}$ *output* (g_1, g_2, Λ, x) *on input* 1^κ. *The* Λ-PSDL assumption *relative to* $\mathrm{C_{gen}}$ *states that* PG_a *is* Λ*-PSDL secure if for any PPT adversary* $\mathscr{A}$,

$$Pr\left[(\mathsf{param_{bp}}) \leftarrow \mathsf{PG}_a(1^\kappa), (g_1, g_2, \Lambda, x) \leftarrow \mathrm{C_{gen}}(1^\kappa),\ x \leftarrow \mathscr{A}\left(\mathsf{param_{bp}}, \Lambda, \left(g_1^{x^i},\ g_2^{x^i}\right)_{i\in\{0\}\cup\Lambda}\right)\right]$$

is negligible in κ.

An alternative definition exists by restricting the input to $\mathsf{param_{bp}}$ and $\left(g_t^{x^i}\right)_{i\in\{0\}\cup\Lambda}$, where $t \in \{1,2\}$. In this case, it is said that PG_a is Λ-PDL secure in $\mathbb{G}_1$ for $t = 1$ (respectively in $\mathbb{G}_2$ for $t = 2$). Furthermore, Lipmaa provided a proof in [Lip12a] that the Λ-PSDL assumption holds in the *generic group model* for any $\Lambda = \{\lambda_i\}_{i\in\mathbb{Z}_n} \subset \mathbb{N}$ such that $\forall i < j$: $0 < \lambda_i < \lambda_j$. This last model is a security model that idealizes reality. Its idealization provides to all participants, a random encoding of a group, together with the corresponding oracles for group operations. It should nevertheless be mentioned that this model has been criticized by Dent in [Den02], as it posses problems when constructing concrete instantiations.

The last two problems presented hereafter are used in related prior protocols ([DJ01, dMW06]), and are explained for purposes of clarity and comparison.

Assume the following definitions: a *literal* is either a boolean variable or its negation; a *clause* is an expression of literals linked with disjunctions (OR: $\vee$); and a *boolean expression* is in the conjunctive normal form (CNF), if it is an expression of clauses linked with conjunctions (AND: $\wedge$).

Decisional Hard Problem 2.12 (Boolean 3-Satisfiability (3SAT) Problem)
Given a boolean expression in the CNF with at most three literals in each clause, decide if it is satisfiable.

The unproven *exponential time hypothesis* [IP99] implies that the 3SAT problem cannot be solved in time faster than $e^{o(n)}$, where n is the number of variables in the corresponding boolean expression.

2.2.3 Random Oracle Model and Hash Functions

Some cryptographic protocols request the usage of *random functions*. These functions output values that are truly random in function of the input. Unfortunately, in order to do so, they require a true randomness source which renders them unpractical. Thus, *hash functions* are used in their stead. Hash functions are deterministic functions that should meet two conditions: they need to be efficiently computable in polynomial time; and it should be hard to find two different inputs that would induce the same output. Their aim is to be hard to invert as well as hard to predict. Nevertheless, they raise concerns when proving security, as their output is deterministic and not truly random.

To provide some security evidence, an extended model was presented in [BR93] by Bellare and Rogaway: the *random oracle model*. In this idealized model, security is proven by replacing hash functions with idealized random functions available to all parties, which are called *random oracles*. A random oracle returns a truly random value when queried with a new input, otherwise it returns the corresponding previously randomly drawn value. Thus, security is proven by considering the underlying hash functions as ideal random functions. This idealization led to several critiques from Canetti, Goldreich, and Halevi in [CGH98, CGH04], and from Goldwasser and Kalai in [GK03]. They showed that some protocols are secure in the random oracle model but insecure under any instantiations of hash functions. These critiques have been minimized by Bleumer in [vTJ11], by pointing out that these cases are usually avoided in the cryptographic literature. Counterbalancing the arguments of Bleumer, Bitansky et al. gave more criticism against the random oracle model in [BDG+13]. They showed that proving security for presumably safer protocols (*statistically sound proofs*, see Section 2.3.1) can also be problematic in the standard model. The random oracle model is nevertheless often preferred in the construction of some specific protocols, such as *non-interactive zero-knowledge proofs* (see Section 2.3.6), due to its practicality when deploying corresponding protocols. In particular, the random oracle model does not require any trusted third party, and often protocols constructed in this model are more efficient. For these reasons, this model is frequently relied on by some practitioners to construct their protocols.

Hash functions in practice take arbitrary length inputs and output fixed length results. Hash functions can either be instantiated with a key (they are thus called *keyed hash functions*) or without (*unkeyed hash functions*).

Although unkeyed hash functions are preferably used, keyed hash functions are used to produce ***Message Authentication Code* (MAC)** algorithms. A MAC algorithm solves the problem of authenticating a large message over an insecure channel. This problem also includes checking the integrity of messages. The solution provided consists of sharing a secret common key between the sender and the receiver, and using this key to produce a small hash digest (called a MAC value or simply a MAC) that will be appended to the message. Informally, the security is based on the difficulty of recovering the secret key, as well as the difficulty of producing a MAC value without the knowledge of the secret key.

Regarding unkeyed hash functions, three security properties need to be presented.

(First) Preimage Resistance: Given a hash digest y, finding x such that $y = h(x)$ must be computationally infeasible.

Second Preimage Resistance: Given a message x and its corresponding hash digest $h(x)$, finding a second different message $x' \neq x$ such that $h(x) = h(x')$ must be computationally infeasible.

Collision Resistance: Finding two different messages $x \neq x'$, such that $h(x) = h(x')$ must be computationally infeasible.

Hash functions that satisfy both the first and second preimage resistance security properties, are called ***one-way*** *hash functions*. If they additionally satisfy the collision resistance security property, they are then called ***collision resistant*** *(or collision intractable) hash functions*. Under certain conditions detailed by Rogaway and Shrimpton in [RS04], the collision resistance property implies both the first and second preimage resistance properties.

One of the most important standards regarding hash functions, is provided by the National Institute of Standards and Technology (NIST) with the ***Secure Hash Algorithm*** **(SHA)** family of hash functions ([NIST15a, NIST15b]). The latest update of the SHA family is referred to as SHA-3 [NIST15b]. This update is the result of a public competition won by Bertoni et al. with their hash function named *Keccak* [BDPA13]. At present, it is not meant to replace the preceding update SHA-2 [NIST15a], and it is considered an alternative standard. Note that SHA-512 corresponds to the hash function in SHA-2 that outputs 512-bit hash digests. Although available in [NIST15a, NIST15b], details of the design of the SHA family are not relevant for this thesis.

2.2.4 Common Reference String Model

In some settings a trusted third party is necessary at the beginning of cryptographic protocols, to provide a trusted common setup for the parameters of protocols. This happens for instance in the case where participants distrust each others but still need some trusted common parameters. To cover this necessity, the Common Reference String (CRS) model [BFM88] provides all participants with the same setup parameters. These parameters can take various forms, such as a uniformly random bit string, or descriptions of parameters (public keys with discarded secret keys, bit lengths, distributions, functions, elements, or sets taken from public distributions, ...). These common setup parameters form a set that is called the *common reference string* (crs). The main application of the CRS model appears in the construction of *non-interactive zero-knowledge proofs*, as explained in Section 2.3.6. In situations where a higher level of trust in cryptography is needed, and participants cannot rely on unproven heuristics such as in the random oracle model, the CRS model offers a good alternative, although it requires a trusted third party to generate the crs. The CRS model will thus be used

to construct the non-interactive protocols presented in Chapter 5. Although this model allows the achievement of efficient results once participants have access to the crs, it is important to keep in mind that the crs itself needs to be transmitted. Thus its size should be kept as small as possible.

2.2.5 Knowledge Assumptions

Knowledge assumptions, which should not be confused with proofs of knowledge (see Section 2.3.3), define the inference between the ability to perform some specific computations (without interaction) and the knowledge required to perform these computations. Informally, assume that some computations Γ can be performed solely with the help of a secret element sk. Then the knowledge assumption states that if the adversary can compute Γ, then it can be inferred that the adversary "knows" sk.

The first knowledge assumption was introduced in 1991 by Damgård in [Dam91]. In 2008, Canetti and Dakdouk gave, in [CD08], a first abstraction of knowledge assumption with the notion of *extractable functions*, where the extractability property of these function ensures the knowledge assumption. Extractable functions are either one-way or collision resistant functions (similarly to hash functions). Furthermore, they assume the existence of an extractor $X_{\mathscr{A}}$ defined for a PPT ITM $\mathscr{A}$ (see Section 2.3.1), such that when given the same inputs (including the random tape) and code of $\mathscr{A}$, $X_{\mathscr{A}}$ outputs a preimage.

Although some specific knowledge assumptions are well defined and accepted, the general definition of a knowledge assumption has not yet been formally set and is still debated within the cryptographic community [GS12b, BCPR14]. The following definition attempts to provide some indications as to how to define knowledge assumptions, and should not be considered a formally matured definition.

Definition 2.20 (Knowledge Assumption)
Assume that $\mathscr{A}$ is a PPT ITM with[1] *auxiliary input x and random tape r. The length of the auxiliary input x is restricted to a polynomial in the security parameter κ (see [BCPR14] for more details), and is either partially or completely generated by the algorithm of the challenger $\mathrm{C}_{\mathrm{gen}}$. Moreover, x is sometimes referred to as the* auxiliary information. *A knowledge assumption is defined with respect to an extractable one-way function f, and states that for any $\mathscr{A}(x,r)$ that achieves the computation of f and outputs y in the image of f, there exists an extractor $X_{\mathscr{A}}(x,r,y)$ that takes as input (x,r,y) and outputs a preimage of y with probability $\big(1-\mu(\|x\|)\big)$, where μ is a negligible function in the length of x.*

Note that in [GS12b], f is called a *Knowledge Commitment Protocol* instead of an extractable function. Lastly, both [GS12b] and [BCPR14] noted that some restrictions need to be added

[1] The exact quantification of x and r depends on the specific knowledge assumption.

to this latter definition, regarding the length and the nature of the auxiliary input x. Impossibility results occur when these restrictions are not made. Nevertheless, several knowledge assumptions were proven secure in the *generic group model*. Recall that this last model is an idealization of reality, as is the random oracle model, and that it has been criticized by Dent in [Den02] for having the same problems as the random oracle model.

This overview on knowledge assumptions aims to introduce the *Λ-Power Knowledge of Exponent* (Λ-PKE) assumption that will be used in Chapter 5, and which was initially formalized in [Lip12a]. Given the parameters of an asymmetric bilinear pairing $\mathsf{param}_{\mathsf{bp}}$ generated by an asymmetric pairing generator $\mathsf{PG}_a(1^\kappa)$, the Λ-PKE security is defined for either $\mathbb{G}_1$ or $\mathbb{G}_2$. Furthermore, assume that the corresponding generator is g, that $(\widehat{\alpha},\ x)$ is a pair of secret parameters, and that $\Lambda = \{\lambda_i\}_{i\in\mathbb{Z}_n} \subset \mathbb{N}$ such that $\forall i < j\colon 0 < \lambda_i < \lambda_j$. The Λ-PKE then informally states that given $\left(g^{x^i},\ g^{\widehat{\alpha}x^i}\right)_{i\in\{0\}\cup\Lambda}$, it is infeasible to produce a pair $(c,\ \widehat{c})$ such that $\widehat{c} = c^{\widehat{\alpha}}$, without knowing variables $(a_i)_{i\in\{0\}\cup\Lambda}$ such that $c = \prod_{i\in\{0\}\cup\Lambda}\left(g^{x^i}\right)^{a_i}$. Assume that g_1 and g_2 are respectively generators of $\mathbb{G}_1$ and $\mathbb{G}_2$.

Knowledge Assumption 2.1 (Λ-PKE [Lip12a])
Let $t \in \{1,2\}$ determine which group ($\mathbb{G}_1$ or $\mathbb{G}_2$) the Λ-PKE security is defined for. Let PG_a output $(\mathsf{param}_{\mathsf{bp}})$ on input (1^κ) and $\mathrm{C}_{\mathrm{gen}}$ output $(g_t, \Lambda, \widehat{\alpha}, x)$ on input $(\mathsf{param}_{\mathsf{bp}}, 1^\kappa)$. The Λ-PKE assumption relative to $\mathrm{C}_{\mathrm{gen}}$, states that PG_a is Λ-PKE secure in $\mathbb{G}_1$ for $t = 1$ (respectively in $\mathbb{G}_2$ for $t = 2$), if for any PPT adversary $\mathscr{A}$ there exists a PPT extractor algorithm $X_{\mathscr{A}}$ such that

$$Pr\left[\begin{array}{c} (\mathsf{param}_{\mathsf{bp}}) \leftarrow \mathsf{PG}_a(1^\kappa),\quad (g_t, \Lambda, \widehat{\alpha}, x) \leftarrow \mathrm{C}_{\mathrm{gen}}\left(\mathsf{param}_{\mathsf{bp}}, 1^\kappa\right), \\ \left((c,\ \widehat{c})\ ;\ (a_i)_{i\in\{0\}\cup\Lambda}\right) \leftarrow (\mathscr{A} \parallel X_{\mathscr{A}})\left(\mathsf{param}_{\mathsf{bp}},\ \left(g_t^{x^i},\ g_t^{\widehat{\alpha}x^i}\right)_{i\in\{0\}\cup\Lambda}\right) : \\ \widehat{c} = c^{\widehat{\alpha}} \ \wedge\ c \neq \prod_{i\in\{0\}\cup\Lambda}\left(g_t^{x^i}\right)^{a_i} \end{array}\right] \text{ is negligible in } \kappa,$$

where the notation $\left(y\ ;\ z\right) \leftarrow (\mathscr{A} \parallel X_{\mathscr{A}})(\chi)$ means that with the same random tape and on input χ, the adversary $\mathscr{A}$ outputs y and the extractor $X_{\mathscr{A}}$ outputs z.

Moreover, the Λ-PKE assumption is derived from the q-PKE assumption of Groth [Gro10], where Λ is replaced by the set $\{1,\ \ldots,\ q\}$. Groth proved in [Gro10] that the q-PKE assumption holds in the generic group model, and his proof naturally extends to the case of Λ. Note that historically, the q-PKE assumption derives from the Knowledge-of-Exponent (KEA3) assumption of Bellare and Palacio [BP04], which itself comes from the Knowledge-of-Exponent (KEA) assumption of Damgård [Dam91]

2.3 Proofs and Arguments

Although ambiguous, *proofs* (or *proof systems*) and *arguments* refer here to some specific protocol constructions, and should not be confused with *security proofs*. A *proof system*

usually involves two types of participants: *provers* and *verifiers*. Sometimes, a trusted third party is also involved. The goal of a proof system is to convince PPT verifiers of the veracity of a statement held by provers. Arguments are a weaker version of proof systems, where provers are limited to being PPT. An additional variant of proof systems, called a *proof of knowledge* (PK), requires provers to know a witness of the statement being proven.

Furthermore, proofs may have a supplementary security property that limits the amount of information that verifiers can obtain. The *zero-knowledge* (ZK) security property states that apart from the veracity of the statement, no information is obtained by verifiers. Weaker notions exist such as *witness hiding* (WH) and *witness indistinguishability* (WI), where the security is focused on how much information verifiers are able to obtain regarding the witness of provers. The last characterization of proofs and arguments targets their interaction model. The latter can be either *interactive* or *non-interactive* depending on whether provers and verifiers are allowed to reply to messages exchanged, or if a single message is sent from the prover to potential verifiers.

Last but not least, arguments are often wrongly called proofs in the cryptographic literature, especially regarding *range proofs* (see chapters 4 and 5). Although this abuse of terminology is also used in the rest of this thesis, the distinction will be preserved in this section.

2.3.1 Interactive Proofs

An *interactive proof protocol*, sometimes called an *interactive proof system*, aims to convince some verifiers of the veracity of a statement to which provers hold some secret evidence. In order to convince verifiers, it is assumed that more than one message is exchanged, hence the interactive designation. If the protocol succeeds to convince verifiers, then it is said that verifiers *accept* the veracity of the statement being proven, or simply that verifiers *accept*. The size of the proof protocol should be relatively small and its verification should be efficient. This is formalized with the complexity class $\mathcal{NP}$.

Definition 2.21 (Complexity Class $\mathcal{NP}$ [Gol01] (Definition 1.3.2, Section 1.3))
A formal language L_R for a relation R is the set of elements x, called words*, that are constrained by the set of rules defined by the relation R. A language L_R is in the complexity class $\mathcal{NP}$ if there exists a boolean relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ and a polynomial p such that R can be recognized in deterministic polynomial time, and $x \in L_R$ if and only if there exists a w such that $\|w\| \leqslant p(\|x\|)$ and $(x,w) \in R$.*

Statements are thus of the form $x \in L_R$, where the common input is x and where provers hold a witness w such that $(x,w) \in R$. Formally, provers and verifiers are defined as *interactive Turing machines* (ITMs), where verifiers are PPT ITMs and provers are unrestricted ITMs. The complete formal definition of an ITM is described by Goldreich in [Gol01], and briefly recalled here.

Definition 2.22 (Interactive Turing Machine [Gol01] (Definition 4.2.1, Section 4.2))
An interactive Turing machine (ITM) is a deterministic multi-tape Turing machine. ITMs are considered in pairs. Moreover, apart from the basic tapes of a Turing machine (read-only input tape, read-only random tape, read-and-write work tape, and write-only output tape), ITMs additionally have two communication tapes (read-only input communication tape, and write-only output communication tape) and an active/idle indication tape (read-and-write switch tape). The additional tapes model the interaction process. The input communication tape of an ITM corresponds to the output communication tape of the other ITM. The switch tape determines which ITM is active and which is idle.

Lastly, the security proof of interactive proof protocols is obtained from two security conditions: *completeness* and *soundness*. The completeness condition states that for any honest prover and any honest verifier, if $x \in L_R$, verifiers accepts the veracity of the statement being proven with a probability of at least $\mathscr{C}$. In general, protocols with perfect completeness are preferred, which implies that $\mathscr{C} = 1$. Informally, this means that the protocol will always succeed with honest interactions. The soundness condition states that for any (potentially malicious) prover and any honest verifier, if $x \notin L_R$, verifiers will accept the veracity of the statement being proven with probability at most $\mathscr{S}$, where $\mathscr{S}$ is ideally negligible. This condition prevents any malicious provers from convincing verifiers of a false statement. Note that the probabilities $\mathscr{C}$ and $\mathscr{S}$ are called, respectively, the completeness error and the soundness error. They are defined by the means of functions in the length of x or equivalently in the security parameter κ, and are respectively named completeness and soundness bounds. The formal definition of an interactive proof can be found in [Gol01], in the definition of "*Generalized Interactive Proof*" (Definition 4.2.6, Section 4.2). The following is an informal reminder of that definition, where the existence of a setup algorithm, in charge of generating the inputs of P and V, will be implicitly assumed.

Definition 2.23 (Interactive Proof)
Assume that a generator algorithm $\mathrm{P}_{\mathrm{gen}}(1^{\kappa})$ *selects a pair* (x, w) *in the relation* R. *An* interactive proof *for language* L_R, *is the composition of a verifier* V *and a prover* P, *such that* V *is PPT ITM with input* x *(bounded in length by the security parameter* κ*),* P *is an unrestricted ITM with provided input* $(x, w) \in R$, *and where the completeness and soundness conditions with respect to* L_R *hold. The interactive proof is thus denoted* $(P, V)^{L_R}$ *or simply* (P, V) *if* L_R *is clear from the context.*

2.3.2 Interactive Argument

Interactive arguments, also known as *computationally sound proof systems*, are interactive proof protocols where malicious provers in the soundness notion are restricted to being PPT ITMs. This restriction also applies to adversaries. Therefore, the (computational) soundness is defined for all (including malicious) PPT provers, where cheating is feasible with negligible

probability in the security parameter κ. In general, the security is based on computational hardness assumptions, as these assumptions assume PPT adversaries.

2.3.3 Proofs of Knowledge (PK)

The definition of *proofs of knowledge* was initiated by the work of Feige, Fiat, and Shamir in [FFS88]. It was then developed by the work of Bellare and Goldreich in [BG92], and extended by Goldreich in [Gol01]. Informally, a *proof of knowledge* is an interactive proof such that the soundness condition additionally requires that the verifier V accepts only if the prover P "knows" $x \in L_R$. This concept of knowledge is captured with the help of a PPT ITM, called the *knowledge extractor* K, such that when given access to a successful prover P, K outputs a witness for $x \in L_R$. Note that in the cryptographic literature regarding proofs of knowledge, P is often assumed to be PPT. Although the term *arguments of knowledge* would be more suitable, these protocols are still referred to as proofs of knowledge.

Definition 2.24 (Proof of Knowledge [Gol01] (Definition 4.7.2, Section 4.7.1))
A proof of knowledge *is an interactive argument that has the following extended soundness condition with a* knowledge error[2] $\mu(\kappa)$.

Denote x as the common input. For the prover P, denote y as its auxiliary input, and r as its random tape. There exists a polynomial q and a probabilistic oracle machine K such that for every prover P, every $x \in L_R$, and every $y, r \in \{0,1\}^*$, machine K satisfies the following condition:

> *Denote by $\mathscr{S}$ the probability that the verifier V accepts, on input x, when interacting with the prover P. If $\mathscr{S} > \mu(\kappa)$, then, on input x and with oracle access to P, machine K outputs a witness w for $x \in L_R$ within an expected number of steps bounded by $\frac{q(\kappa)}{\mathscr{S}-\mu(\kappa)}$.*

The oracle machine K is called a universal knowledge extractor*, or more simply a* knowledge extractor. *Proofs of knowledge will be denoted $PK\{(w,r) : x \in L_R\}$, where the elements (w,r) represent the elements the knowledge of which is being proved, and $x \in L_R$ the statement related to the proof of knowledge.*

Furthermore, note the existence of some alternative (weaker) definitions. For instance, the running time of the extractor is commonly assumed to run in expected polynomial-time instead of strict polynomial-time. Another notable example is the definition from [Lin01, Lin03] where the soundness condition is replaced with a weaker version called a *witness-extended emulation*. Informally, in this latter definition, the extractor K is replaced with an emulator running in expected polynomial-time (and not PPT), which outputs an emulation of

[2] The knowledge error μ is often associated to the maximum probability with which a verifier V can be convinced on input $x \notin L_R$.

a successful proof produced by the adversary, together with a witness corresponding to the emulation.

2.3.4 Zero-Knowledge (ZK)

Introduced by Goldwasser, Micali, and Rackof in [GMR85, GMR89], *zero-knowledge* is an important security property for proofs, arguments, and proofs of knowledge. This property aims to protect provers against malicious verifiers which attempt to gain more knowledge than that which is intended. Informally, the goal of ZK is that verifiers learn "no" knowledge besides that which is intended, even when they deviate from the protocol execution. Regarding a proof protocol where the statement $x \in L_R$ is being proven, ZK guarantees that verifiers will only learn the veracity of the statement, without any additional knowledge, hence the term zero-knowledge. This security property is achieved by showing that the interaction with the prover can be efficiently simulated without the help of the prover, and thus without his private inputs and without the witness w of $x \in L_R$. This simulation is performed by a PPT ITM, called the *simulator*. Furthermore, the collection of messages exchanged in a proof protocol is called the *transcript* of the proof protocol. Hence, the goal of the simulator is to simulate the transcript.

Definition 2.25 (Zero-Knowledge)
Let $(P, V)^{L_R}$ be a proof protocol regarding the language L_R, between a prover P and a verifier V. Let $tr(P, V^*)^{L_R}$ be the sequence of random variables defining the transcript distribution between any verifier V^* and the prover P. Let tr_{sim} be the sequence of random variables defining the simulated transcript distribution, where the simulated transcript is output by the simulator. Note that the sole input of the simulator is the common input x.

The zero-knowledge *property of $(P, V)^{L_R}$ depends on the indistinguishability relation between $tr(P, V^*)^{L_R}$ and tr_{sim}. Thus the proof protocol $(P, V)^{L_R}$ is said to be:*

- perfectly zero-knowledge *if for all $x \in L_R$,*
  $tr(P, V^*)^{L_R}$ and tr_{sim} are perfectly indistinguishable $\left(tr(P, V^*)^{L_R} =_p tr_{sim}\right)$.
- statistically zero-knowledge *if for all $x \in L_R$,*
  $tr(P, V^*)^{L_R}$ and tr_{sim} are statistically indistinguishable $\left(tr(P, V^*)^{L_R} =_s tr_{sim}\right)$.
- computationally zero-knowledge *if for all $x \in L_R$,*
  $tr(P, V^*)^{L_R}$ and tr_{sim} are computationally indistinguishable $\left(tr(P, V^*)^{L_R} =_c tr_{sim}\right)$.

Recall that the indistinguishability relation is explained in Section 2.1.2. This definition of zero-knowledge is applicable to arguments as well as to proofs of knowledge, by considering the prover P as a PPT ITM. Furthermore, if the definition is restricted to honest verifiers (V^* is restricted to honest verifiers), then the security property achieved is called honest verifier zero-knowledge (HVZK). It can also be noted that more diverse forms of zero-knowledge exist (see [Gol01], Chapter 4).

2.3.5 Weaker ZK with witness security

Feige and Shamir introduced in [FS90] two weaker versions of zero-knowledge, where security focuses on the witness of the statement. The first one, called *witness hiding* (WH), states that (cheating) verifiers are unable to compute a witness for the statement, even after interacting with the honest prover, unless verifiers were able to do so prior to any interactions with the honest prover. The second security property, called *witness indistinguishability* (WI), states that (cheating) verifiers are unable to identify which witnesses are held by the prover, even if all witnesses are known to the verifiers. Furthermore, Feige and Shamir also showed in [FS90] that WH can be instantiated from WI protocols.

Definition 2.26 (Witness Indistinguishability [FS90] (Definition 3.1, Section 3))
Let $tr\left(P_{(x,w)},\ V_{(x,\widetilde{y})}\right)^{L_R}$ *be the sequence of random variables defining the transcript distribution of a proof protocol* $(P,\ V)^{L_R}$*, where* x *is the common input,* w *is the witness of the prover* P *and* $\widetilde{y}$ *is the auxiliary input that the verifier* V *might have.*

$(P,\ V)^{L_R}$ *is* witness indistinguishable *(WI) regarding* L_R*, if for any (potentially malicious) verifier* V^**, for any long enough common input* x*, for any witnesses* w_1, w_2 *such that* $(x, w_1) \in R$ *and* $(x, w_2) \in R$*, the following holds:* $tr\left(P_{(x,w_1)},\ V^*_{(x,\widetilde{y})}\right)^{L_R}$ *and* $tr\left(P_{(x,w_2)},\ V^*_{(x,\widetilde{y})}\right)^{L_R}$ *are perfectly indistinguishable.*

WI is sometimes referred to as *perfect witness indistinguishability* to differentiate it with the statistical case, where $tr\left(P_{(x,w_1)},\ V^*_{(x,\widetilde{y})}\right)^{L_R}$ and $tr\left(P_{(x,w_2)},\ V^*_{(x,\widetilde{y})}\right)^{L_R}$ are statistically indistinguishable.

2.3.6 Non-Interactive (NI) Proofs

Interactive proofs (as well as arguments and PK) also have a *non-interactive* (NI) variant. The essential difference with non-interactive proofs is that all interactions consist of a single message sent by the prover to the verifier. Verifiers are thus passive in the sense that they send no messages to provers. NI proofs were initially introduced in the context of non-interactive zero-knowledge (NIZK) proofs [BFM88, SMP87, BSMP91]. Furthermore, it has been shown in [GO94] that NIZK proofs require some setup assumptions, such as provided with the CRS or with the random oracle model. Therefore, the standard model alone is insufficient for NIZK proofs. Moreover, zero-knowledge for NIZK proofs, in the context of the CRS model, is achieved if both the CRS and the proof can be simulated.

Definition 2.27 (NIZK Proofs)
A non-interactive argument *for language* L_R *consists of the next PPT algorithms: a CRS generator* $\mathsf{Gen}_{\mathsf{crs}}$*, a prover* P*, and a verifier* V*. For* $\mathsf{crs} \leftarrow \mathsf{Gen}_{\mathsf{crs}}(1^\kappa)$*,* $P(\mathsf{crs}; x, w)$ *produces an argument* π*. The verifier* $V(\mathsf{crs}; x, \pi)$ *outputs either* 1 *(accept) or* 0 *(reject).*

A non-interactive argument $(\mathrm{Gen}_{\mathsf{crs}}, P, V)$ *is* perfectly complete*, if for all* $\mathsf{crs} \leftarrow \mathrm{Gen}_{\mathsf{crs}}(1^\kappa)$ *and all* $(x,w) \in R$*,* $V(\mathsf{crs}; x, P(\mathsf{crs}; x, w))$ *outputs* 1*. A non-interactive argument* $(\mathrm{Gen}_{\mathsf{crs}}, P, V)$ *is* computationally (adaptively) sound*, if for all non-uniform PPT adversaries* $\mathscr{A}$*,*

$$\Pr\left[\mathsf{crs} \leftarrow \mathrm{Gen}_{\mathsf{crs}}(1^\kappa), (x,\pi) \leftarrow \mathscr{A}(\mathsf{crs}) : x \notin L_R \wedge V(\mathsf{crs}; x, \pi) \text{ outputs } 1\right]$$

is negligible in κ*.*

A non-interactive argument $(\mathrm{Gen}_{\mathsf{crs}}, P, V)$ *is* perfectly witness indistinguishable*, if (given that there are several possible witnesses) it is impossible to tell which witness the prover used. That is, if* $\mathsf{crs} \leftarrow \mathrm{Gen}_{\mathsf{crs}}(1^\kappa)$ *and* $((x,w_0),(x,w_1)) \in R^2$*, then the distributions* $P(\mathsf{crs}; x, w_0)$ *and* $P(\mathsf{crs}; x, w_1)$ *are equal.*

$(\mathrm{Gen}_{\mathsf{crs}}, P, V)$ *is* perfectly zero-knowledge*, if there exists a polynomial-time simulator* $Sim = (Sim_1, Sim_2)$ *with a* simulation trapdoor td*, such that for all stateful interactive non-uniform PPT adversaries* $\mathscr{A}$*,*

$$\begin{aligned}&\Pr\left[\begin{array}{c}\mathsf{crs} \leftarrow \mathrm{Gen}_{\mathsf{crs}}(1^\kappa), (x,w) \leftarrow \mathscr{A}(\mathsf{crs}), \pi \leftarrow P(\mathsf{crs}; x, w) : \\ (x,w) \in R \wedge \mathscr{A}(\mathsf{crs}, \pi) \text{ outputs } 1\end{array}\right] \\ &= \Pr\left[\begin{array}{c}(\mathsf{crs}, \mathsf{td}) \leftarrow Sim_1(1^\kappa), (x,w) \leftarrow \mathscr{A}(\mathsf{crs}), \pi \leftarrow Sim_2(\mathsf{crs}, x, \mathsf{td}) : \\ (x,w) \in R \wedge \mathscr{A}(\mathsf{crs}, \pi) \text{ outputs } 1\end{array}\right].\end{aligned}$$

$(\mathrm{Gen}_{\mathsf{crs}}, P, V)$ *is* computationally zero-knowledge *if these two probabilities are computationally indistinguishable.*

Definition 2.28 (NIZK Proofs of Knowledge)
A non-interactive zero-knowledge proof of knowledge *is a non-interactive zero-knowledge argument that has the following extended soundness condition with a* knowledge error $\mu(\kappa)$*.*

Let crs *be generated by the CRS generator* $\mathrm{Gen}_{\mathsf{crs}}(1^\kappa)$*. The non-uniform PPT adversary* $\mathscr{A}$ *is given* crs *as input, and outputs* x *(the common input) together with its corresponding proof* π*. Denote* $\mathscr{S}$ *as the probability that a verifier* V *accepts on input* $(\mathsf{crs}; x, \pi)$*. The knowledge extractor* $X_{\mathscr{A}}$ *is a probabilistic oracle machine that takes as input* $x \in L_R$*, the* crs*, and the random tape of* $\mathscr{A}$*.*

The extended soundness condition holds if there exists a polynomial q*, and a knowledge extractor* $X_{\mathscr{A}}$*, such that if* $\mathscr{S} > \mu$*, then* $X_{\mathscr{A}}$ *outputs a witness* w *for* $x \in L_R$ *within an expected number of steps bounded by* $\frac{q(\kappa)}{\mathscr{S} - \mu(\kappa)}$*.*

Identically to proofs of knowledge, non-interactive zero-knowledge proofs of knowledge will be denoted $NIZK\text{-}PK\{(w,r) : x \in L_R\}$*.*

Since their introduction, the importance of NIZK proofs rose and they are now considered an important part of cryptography. NIZK proofs can be either constructed directly in the CRS

model (by using, for instance, a general methodology), or by converting an interactive equivalent proof using generic transformations. Notable examples of such generic transformations, are the use of the Fiat-Shamir heuristic [FS86], the Lindell transform [Lin15], and the CPSV transform [CPSV16]. These three transformations are restricted to particular interactive proofs called Σ-protocols (see Section 2.4.1) and will result in a security proven in the random oracle model. Although the Fiat-Shamir heuristic yields very efficient protocols, it is considered controversial as some insecurity results were shown in [CGH98, CGH04]. Alternatively, the Groth-Sahai method [GS08, GS12a] is the most popular general methodology for obtaining a NIZK proof in the CRS model, even though this method is focused on protocols based on bilinear pairings. Nevertheless, both the random oracle model and the CRS model will be considered in this thesis for the sake of completeness. This will allow users to select their protocols according to the security model they need.

2.4 Protocols and Building Blocks

This section explains the majority of the protocols and building blocks that are needed to understand the various results of this thesis. The set membership proofs of Chapter 3 and the interactive range proofs of Chapter 4 are all based on Σ-protocols, which are explained in Section 2.4.1. Commitment schemes, public key cryptosystems, signature schemes, and accumulators are respectively defined in sections 2.4.2, 2.4.3, 2.4.5, and 2.4.6, together with the specific instantiations needed for this thesis. Lastly, the non-interactive range proof of Chapter 5 requires the lifted BBS cryptosystem, the Hadamard product argument and the Lipmaa permutation argument, which are respectively explained in sections 2.4.4, 2.4.7, and 2.4.8.

2.4.1 Σ-Protocols

Σ-protocols are interactive zero-knowledge proofs (or arguments) of knowledge that follow a specific interaction pattern with special security properties. The interaction pattern consists of three messages (a, c, r). The first message a is sent by the prover to the verifier and is usually a commitment to the randomness of the prover. The second message c is a sufficiently large random challenge, sent by the verifier to the prover. The last message r is a specific response from the prover, that allows the verifier to achieve the proof of knowledge by running a deterministic check on his inputs and the messages exchanged.

The special security properties that need to be met are *perfect completeness*, *special soundness*, and *special honest verifier zero-knowledge* (SHVZK). The perfect completeness property is identical to the one for proofs of knowledge. The special soundness property only restricts the oracle access to the prover for the knowledge extractor. For any $x \in L_R$ and on any pair of accepting interactions (a, c, r) and (a, c', r') such that $c \neq c'$, the knowledge extractor

tries to output the witness w, given as input (x, a, c, c', r, r'). The SHVZK property implies the HVZK property for a transcript of the form (a, c, r), with the additional property that, given a uniformly chosen challenge c, the simulator outputs an accepting interaction (a, c, r). Moreover, computationally indistinguishability of the simulated transcript is usually assumed in SHVZK.

Definition 2.29 (Σ-Protocols)
A Σ-protocol for language L_R is an interactive zero-knowledge proofs (or arguments) of knowledge (P, V) where the transcript (or conversation) is of the form (a, c, r). The messages a and r are computed by P, and c is a challenge randomly chosen by V. The verifier accepts if $\phi(x, a, c, z) = 1$ for some efficiently computable predicate ϕ, where x is the common input of P and V.

A Σ-protocol must satisfy three security requirements: completeness, special soundness and special honest verifier zero-knowledge (SHVZK). A Σ-protocol is perfectly complete when a honest prover convinces honest verifiers with probability 1. A Σ-protocol has the special soundness property when the knowledge extractor K can efficiently recover a witness w such that $(x, w) \in R$, from two accepting transcripts[3] (a, c, r) and (a, c', r'), where $c \neq c'$. A Σ-protocol has the SHVZK property if there exists a PPT simulator Sim that can first randomly pick c^*, then find r^* and a^* such that the transcript (a^*, c^*, r^*) is accepting and the distribution (a^*, c^*, r^*) is computationally indistinguishable from the distribution of accepting transcripts between honest provers and honest verifiers. Finding r^* and a^* is usually achieved by (randomly) picking r^* first, and deducing a^* after.

The terminology of Σ-protocols was introduced by Cramer in [Cra97], although this type of protocols was used before (as early as in [FS86]). In [CDS94], Cramer et al. showed that the special soundness imply the standard soundness of proofs of knowledge. Furthermore, it is possible to convert a Σ-protocol that is SHVZK to plain ZK as shown by Cramer, Damgård, and MacKenzie in [CDM00].

2.4.2 Commitment Schemes

A *commitment scheme* is one of the basic cryptographic primitives. The concept of commitment schemes was introduced by Blum in [Blu81] and its terminology by Even in [Eve81], although they were implicitly used by Shamir et al. in [SRA81].

Informally, a *commitment* can be seen as a vault with a safe deposit box. A party, called the *committer* will put his message into the safe and lock it. He will then be able to transmit the vault with the assurance that it is safely hidden in the vault. Meanwhile, the other parties will want to have the insurance that the committer will not be able to cheat by providing a wrong opening sequence to another hidden safe box inside the vault, containing a message

[3] Recall that K has a rewindable oracle access to P.

that is different to the initial one. Therefore there is a need to define two conflicting security properties of commitments, the *hiding* and the *binding* requirements.

Thus, a *commitment scheme* is a protocol enabling the *committer* to *commit* to a message of his choice. This means that the committer will be able to fix his decision upon a message and disclose that he has done so without revealing the content of his message. He will be tied to his decision through a mathematical object called *the commitment* linked to his hidden message. In a later phase, when the committer will be asked to reveal his message, the other parties, called *verifiers*, will have the means to verify that his revealed message is indeed unconditionally linked to his commitment. The *hiding* requirement prevents verifiers from learning the content of the commitment. The *binding* requirement prevents the committer to cheat when opening his commitment. Here, cheating means opening the commitment to a different value than the initial message committed in the commitment. The following definitions are inspired by the definitions of Goldreich in [Gol01].

Definition 2.30 (Commitment Scheme)
A (non-interactive) commitment scheme *C is the combination of three algorithms:* Gen*,* Commit*, and* Open*, representing respectively the parameters generation, the commit, and the open algorithm. The* Gen(1^κ) *algorithm generates parameters p for the scheme based on a given security parameter* κ*. The commit algorithm* Commit(p,m,r) *runs on input* (p,m,r) *where m is a message string taken in the message space* $\mathbb{M}$ *and r is a random tape.* Commit *produces a pair of values* (c,o) *representing respectively the* commitment *as a committed string and an opening string. For simplicity, the sub-algorithm that produces the commitment c in the commit algorithm* Commit*, will be denoted* Com(p,m,r) *or* Com(m,r) *if p is clear from the context. The open algorithm* Open(c,m,o) *runs on input* (c,m,o) *and outputs* 1 *or* 0*, whether the commitment c successfully opens to the message m or not.*

Once the commitment parameters have been generated by Gen, the committer $\mathscr{C}$ will run Commit and transmit the commitment c to potential verifiers $\mathscr{V}$. In order to open the commitment, the committer $\mathscr{C}$ just needs to transmit the initial message m together with the opening string o to the verifiers $\mathscr{V}$, who will run the checking algorithm Open.

The *hiding* security requirement in a commitment scheme refers to the difficulty or impossibility for an adversary to determine the message m from c. It ensures the committer that his commitment will leak no information about his message choice. Depending on the indistinguishability property (see Section 2.1.2) of the commitments, three strengths of hiding commitments exist, namely computationally, statistically, and perfectly hiding commitments. A computationally hiding commitment is secure against a computationally bounded adversary. A statistically hiding commitment will resist against computationally unbounded adversaries. Lastly, there exist no adversaries able to break a perfectly hiding commitment.

Definition 2.31 (Hiding)
For a given commitment scheme C and any security parameter κ*, assume that* c_1 *and* c_2 *are any two commitments on different messages* m_1 *and* m_2 *respectively. The* hiding *property*

Common Input:	$p \leftarrow \text{Gen}(1^\kappa)$.
Prover Input:	(m, r).
	Commit Phase
$\mathscr{C} \xrightarrow{c} \mathscr{V}$	Commiter runs $(c, o) \leftarrow \text{Commit}(p, m, r)$ and sends c.
	Open Phase
$\mathscr{C} \xrightarrow{m,o} \mathscr{V}$	Commiter sends m, o.
	Verifier checks that $\text{Open}(c, m, o) \stackrel{?}{=} 1$.

Protocol 2.1 – General commitment scheme with security parameter κ

states that an adversary $\mathscr{A}$ cannot distinguish a commitment to m_1 from a commitment to m_2. This means that an adversary $\mathscr{A}$ given as input either (κ, p, c_1) or (κ, p, c_2), will have a computationally, statistically, or perfectly indistinguishable output distribution. This indistinguishability relation defines the hiding property type.

Formally, let $(c_i)_\kappa$ denote the sequence of random variables representing the commitment distribution from Com *on input m_i, and indexed with the security parameter κ. The* hiding *property of the commitment scheme C depends on the indistinguishability relation between $\{(c_1)_\kappa\}$ and $\{(c_2)_\kappa\}$. A commitment scheme is said to be:*

- computationally hiding
  if $\{(c_1)_\kappa\}$ and $\{(c_2)_\kappa\}$ are computationally indistinguishable $(\{(c_1)_\kappa\} =_c \{(c_2)_\kappa\})$.
- statistically hiding
  if $\{(c_1)_\kappa\}$ and $\{(c_2)_\kappa\}$ are statistically indistinguishable $(\{(c_1)_\kappa\} =_s \{(c_2)_\kappa\})$.
- perfectly hiding
  if $\{(c_1)_\kappa\}$ and $\{(c_2)_\kappa\}$ are perfectly indistinguishable $\left(\{(c_1)_\kappa\} =_p \{(c_2)_\kappa\}\right)$.

The *binding* security requirement in a commitment scheme refers to the difficulty or impossibility of a committer opening the value of his commitment c to two different messages m_1, m_2. The goal is to ensure that the committer is bound to his initial choice once his commitment is created and published. A computationally binding commitment will bind computationally bounded committers. Lastly, there exists no committer that can open one of his commitments to two different messages, if his commitments are perfectly binding.

Definition 2.32 (Binding)
Denote by $\mathscr{C}^*$ the ITM representing a cheating committer with input parameter p. Denote by β the probability that $\mathscr{C}^*$ is able to output a commitment that he can successfully open into

two different messages $m_1, m_2 \in \mathbb{M}$ with corresponding openings o_1, o_2:

$$\beta = \Pr\left[\begin{array}{l} p \leftarrow \mathrm{Gen}\left(1^{\kappa}\right), (c, m_1, m_2, o_1, o_2) \leftarrow \mathscr{C}^*(p) : \\ m_1 \neq m_2 \ \wedge \ Open(c, m_1, o_1) = 1 \ \wedge \ Open(c, m_2, o_2) = 1 \end{array}\right]$$

A commitment scheme is said to be:

- computationally binding
  if there exists a security parameter $\kappa_0 \in \mathbb{N}$ such that for every $\kappa > \kappa_0$, for every positive polynomial $q(\cdot)$, and for every computationally bounded PPT ITM $\mathscr{C}^*$,
  β is negligible in κ: $\beta < \frac{1}{q(\kappa)}$.
- perfectly binding
  if there exists no ITM $\mathscr{C}^*$ that is able to succeed. Hence $\beta = 0$ for all $\mathscr{C}^*$.

Definition 2.33 (Security of Commitment Schemes)
A commitment scheme *C is said to be secure if it satisfies the* hiding *and* binding *security requirements. The indistinguishability reached in these security requirements will define the security level of the commitment scheme.*

Notice that statistically and perfectly hiding commitments (Definition 2.31) protect against computationally unbounded verifiers, whereas perfectly binding commitments (Definition 2.32) protect against computationally unbounded committers. The following theorem states that any two of these hiding/binding security achievements are mutually exclusive.

Theorem 2.2 (Binding and Hiding Antagonistic Protections)
Perfectly binding commitment schemes cannot be achieved simultaneously with either a statistically or a perfectly hiding security requirement.

Proof (informal)
Suppose that a commitment schemes C achieves perfectly binding security, and either statistically or perfectly hiding security.

Then, when creating a commitment c from message m and random tape r, the perfectly binding requirement states that a computationally unbounded committer cannot open c to another message m'. This implies that Com is injective. Hence a computationally unbounded adversary could compute the corresponding m and r of c, breaking the statistically/perfectly hiding security requirement.

In order to satisfy the statistically/perfectly hiding security requirement, there must exist a r' and $m' \neq m$ that leads to the same commitment c on m. However, in this case, a computationally unbounded committer will also be able to find them, breaking the perfectly binding security requirement. ■

Pedersen Commitment

The *Pedersen commitment*[4] presented in [Ped91] is perfectly hiding and computationally binding. Its computational hardness assumption is based on the *discrete logarithm problem* (DLog problem). In the setup phase of the commitment, the generation algorithm chooses a group G of prime order q such that $\|q\| = \kappa$, where κ is the security parameter. A random element h together with a random generator g is picked in G such as $\log_g h$ is unknown (especially from the committer).

The commitment is of the form $c = g^m h^r$. Here $m, r \in \mathbb{Z}_q$ and computation is performed in G. The value m represents the secret message of the committer, and r a uniformly random number of his choice. To open the commitment, the committer simply reveals m and r, to let the prover check that $c = g^m h^r$.

This scheme is perfectly hiding, as a commitment c could hold a commitment to any value of m. For any $m' \neq m$, the corresponding $r' = r + (m - m') \log_g h \pmod{q}$ yields the same commitment c. Hence it is important to keep the value of $\log_g h$ unknown. The binding property holds as it is computationally infeasible to find $\log_g h$, thanks to the DLog assumption. The Pedersen commitment could also be used without the knowledge of the group order q, in order to introduce more mathematical properties in the exponent, as will be seen later.

Fujisaki-Okamoto Commitment

This commitment was introduced by Fujisaki and Okamoto in [FO98] and refined later by Damgård and Fujisaki in [DF02]. It is a statistically hiding and computationally binding commitment scheme based on the hardness of factorization. The parameters generation algorithm $\text{Gen}(1^\kappa)$ outputs three elements (n, g, h). The first element n, of length κ, is the product of two safe primes. n is called a special RSA modulus. $h \in QR_n$ is picked as a quadratic residue modulo n and g is taken from the group generated by h. It is assumed that given (n, g, h), an estimate on the order of $\mathbb{Z}_n$ can be efficiently computed as $ord(\mathbb{Z}_n) \leqslant 2^B$. The commitment on a message m with randomness $r \in_R \mathbb{Z}_{2^{B+\kappa}}$ is computed as $\text{Com}\big((n, g, h), m, r\big) = g^m h^r \pmod{n}$.

Commitments in the CRS Model

Some additional definitions are provided here for the commitment used in the non-interactive range proof presented in Chapter 5.

A *batch commitment scheme* (Gen, Com) for n elements in a bilinear group consists of two PPT algorithms: a randomized CRS generation algorithm Gen, and a randomized commitment

[4] Note that an earlier version with a slight modification was also present in [CDvdG87].

algorithm Com. Let $t \in \{1,2\}$ determine which group ($\mathbb{G}_1$ or $\mathbb{G}_2$) the batch commitment scheme is defined for. Here, $\mathsf{Gen}^t(1^\kappa, n)$ produces a common reference string crs_t. Depending on the context of the commitment, assume a parameter $d \in \{1,2,3\}$. The commitment algorithm $\mathrm{Com}^t(\mathsf{crs}_t, \boldsymbol{a}, r)$, with $\boldsymbol{a} = (a_1, \ldots, a_n)$, outputs a commitment value A in $\mathbb{G}_1^d$ if $t = 1$ or in $\mathbb{G}_2^d$ if $t = 2$. Furthermore the commitment $\mathrm{Com}^t(\mathsf{crs}_t, \boldsymbol{a}, r)$ is opened by revealing $(\boldsymbol{a}, r)$.

A commitment scheme (Gen, Com) is *computationally binding* in group $\mathbb{G}_1$ (respectively $\mathbb{G}_2$) for $t = 1$ (respectively for $t = 2$), if for every non-uniform PPT adversary $\mathscr{A}$ and positive integer n (polynomial in κ), the probability

$$\Pr\left[\begin{array}{l}\mathsf{crs}_t \leftarrow \mathrm{Gen}^t(1^\kappa, n),\ (\boldsymbol{a_1}, r_1, \boldsymbol{a_2}, r_2) \leftarrow \mathscr{A}(\mathsf{crs}_t) : \\ \boldsymbol{a_1} \neq \boldsymbol{a_2} \ \wedge\ \mathrm{Com}^t(\mathsf{crs}_t, \boldsymbol{a_1}, r_1) = \mathrm{Com}^t(\mathsf{crs}_t, \boldsymbol{a_2}, r_2)\end{array}\right]$$

is negligible in κ. A commitment scheme (Gen, Com) is *perfectly hiding* in group $\mathbb{G}_1$ (respectively $\mathbb{G}_2$) for $t = 1$ (respectively for $t = 2$), if for any positive integer n (polynomial in κ), for $\mathsf{crs}_t \in \mathrm{Gen}^t(1^\kappa, n)$, for any two messages $\boldsymbol{a_1}, \boldsymbol{a_2}$, and for any randomness r_1, r_2, the distributions $\mathrm{Com}^t(\mathsf{crs}_t, \boldsymbol{a_1}, r_1)$ and $\mathrm{Com}^t(\mathsf{crs}_t, \boldsymbol{a_2}, r_2)$ are equal.

A *trapdoor commitment scheme* has three additional efficient algorithms: $\mathsf{Gen}_{\mathsf{td}}$, $\mathsf{Com}_{\mathsf{td}}$, and $\mathsf{Open}_{\mathsf{td}}$. The first algorithm $\mathsf{Gen}_{\mathsf{td}}$ is the trapdoor CRS generation algorithm that takes inputs t, n, and 1^κ, and that outputs a trapdoor td and a common reference string crs^* with the same distribution as $\mathrm{Gen}^t(1^\kappa, n)$. The second algorithm $\mathsf{Com}_{\mathsf{td}}$ is a randomized trapdoor commitment that takes crs^* and a randomizer r as inputs, and outputs the value $\mathrm{Com}^t(\mathsf{crs}^*, \mathbf{0}, r)$. Lastly, $\mathsf{Open}_{\mathsf{td}}$ is the trapdoor opening algorithm that takes crs^*, td, $\boldsymbol{a}$, and r as an input and outputs an r' such that $\mathrm{Com}^t(\mathsf{crs}^*, \mathbf{0}, r) = \mathrm{Com}^t(\mathsf{crs}^*, \boldsymbol{a}, r')$.

Finally, an *extractable commitment scheme* is a commitment scheme (Gen, Com) with an additional extractor $\mathscr{E} = (X_1, X_2)$ such that: $X_1(1^\kappa)$ creates a common reference string crs^* (indistinguishable from the real crs) and a trapdoor td ; and $X_2(\mathsf{crs}^*, \mathsf{td}, A)$ returns $(\boldsymbol{a}, r)$ such that $A = \mathrm{Com}(\mathsf{crs}, \boldsymbol{a}, r)$, given that A is a valid commitment. An extractable commitment scheme can only be computationally hiding.

Groth-Lipmaa Knowledge Commitment

The non-interactive range proof presented in Chapter 5 requires the Groth-Lipmaa knowledge commitment scheme in the CRS model, defined in [Gro10, Lip12a] and described as follows.

CRS generation:

Generate a set $\Lambda = \{\lambda_i\}_{i \in \mathbb{Z}_n} \subset \mathbb{N}$ such that $\forall i < j$: $0 < \lambda_i < \lambda_j$, with n polynomial in κ. Given a bilinear group generator PG, set $\mathsf{param}_{\mathsf{bp}} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathrm{T}, e) \leftarrow \mathsf{PG}(1^\kappa)$. Let $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ be generators, and choose random $\widehat{\alpha}, x \in_R \mathbb{Z}_p$. Fix $t \in \{1,2\}$. The common reference string is $\mathsf{crs}_t = \left(\mathsf{param}_{\mathsf{bp}}, g_t, \widehat{g}_t, \left(g_{t,\lambda_i}, \widehat{g}_{t,\lambda_i}\right)_{i \in \mathbb{Z}_n}\right)$, where $g_{t,\lambda_i} = g_t^{x^{\lambda_i}}$,

and $\widehat{g}_{t,\lambda_i} = g_t^{\widehat{a}x^{\lambda_i}}$.

Commitment: To commit to $\boldsymbol{a} = (a_0, \ldots, a_{n-1}) \in \mathbb{Z}_p^n$, one chooses a random $r \in_R \mathbb{Z}_p$, and computes:

$$\mathrm{Com}^t(\mathsf{crs}_t, \boldsymbol{a}, r) = (g_t^r \cdot \prod_{i=0}^{n-1} g_{t,\lambda_i}^{a_i}, \widehat{g}_t^r \cdot \prod_{i=0}^{n-1} \widehat{g}_{t,\lambda_i}^{a_i}).$$

Let $t = 1$. Fix a commitment key crs_1 that in particular specifies $g_2, \widehat{g}_2 \in \mathbb{G}_2$. A commitment $(A, \widehat{A}) \in \mathbb{G}_1^2$ is *valid*, if $e(A, \widehat{g}_2) = e(\widehat{A}, g_2)$. The case of $t = 2$ is dual.

According to [Lip12a], the Groth-Lipmaa knowledge commitment scheme is statistically hiding and computationally binding in group $\mathbb{G}_1$ for $t = 1$ (or $\mathbb{G}_2$ for $t = 2$) under the Λ-PSDL assumption in the corresponding group. If the Λ-PKE assumption holds in $\mathbb{G}_1$ (respectively $\mathbb{G}_2$), then for any PPT algorithm $\mathscr{A}$ that outputs some valid knowledge commitments, there exists a non-uniform PPT extractor $X_{\mathscr{A}}$ that, given the same inputs as of $\mathscr{A}$ together with the random tape of $\mathscr{A}$, extracts the contents of these commitments. This knowledge commitment scheme is also trapdoor, with the trapdoor being $\mathsf{td} = x$. After using the trapdoor commitment to produce a commitment $A \leftarrow \mathrm{Com}^t(\mathsf{crs}, \mathbf{0}, r) = g_t^r$ for $r \in_R \mathbb{Z}_p$, the committer can open it to $(\boldsymbol{a},\ r - \sum_{i=0}^{n-1} a_i x^{\lambda_i})$ for any $\boldsymbol{a}$.

2.4.3 Public Key Cryptosystems

Also called asymmetric cryptosystems, public key cryptosystems[5] enable anyone to encrypt a message, called *plaintext*, for a particular recipient. However, encrypted messages, called *ciphertexts*, can be decrypted only by the intended recipient. Three algorithms compose public key cryptosystems: a PPT key generator, a PPT encryption, and a deterministic decryption. The key generator produces two keys, a public and a secret one, for each recipients. The public key is made public for anyone who wants to send an encrypted message to their corresponding recipient. The secret key, also called private key, is secretly kept by its owner. The encryption algorithm takes as input a message together with the public key of the intended recipient, and outputs the corresponding ciphertext. The decryption algorithm requires the proper secret key to decrypt a ciphertext into its corresponding message.

Definition 2.34 (Public Key Cryptosystem (PKC))

A public key cryptosystem *is the combination of three algorithms:* Gen, Enc, *and* Dec, *representing respectively the PPT key generator, the PPT encryption, and the deterministic polynomial-time decryption algorithm. The key generator algorithm* $\mathrm{Gen}(1^\kappa)$ *generates a public key* pk *and a secret key* sk *for a specific user* $\mathscr{U}$*, based on a given security parameter* κ*. The encryption algorithm* $\mathrm{Enc}(m, \mathsf{pk})$ *takes as input the message* m *and the public key* pk*.* $\mathrm{Enc}(m, \mathsf{pk})$ *outputs the ciphertext* γ *of* m *encrypted with* pk*. The decryption algorithm* $\mathrm{Dec}(\gamma, \mathsf{sk})$ *takes as input the secret key* sk *and a ciphertext* γ *produced with the public key* pk

[5] The term "cryptosystem" has sometimes ambiguous meaning in the cryptographic literature. In this thesis it will refer to encryption schemes.

corresponding to sk. $\mathrm{Dec}(\gamma, \mathsf{sk})$ *outputs the decryption* m *of* γ. *A public key cryptosystem is said to be correct if for every pair* $(\mathsf{pk}, \mathsf{sk})$ *generated by* Gen,

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

Although several notions of security exist for public key cryptosystems, only *semantic security against chosen plaintext attacks* will be briefly explained hereafter. More details can be found in Chapter 9 of [Vau06] and in Section 5.4 of [Gol04]. Semantic security was introduced by Goldwasser and Micali in [GM84] and captures the notion that ciphertexts reveal no information on their plaintext. The latter notion is illustrated by the indistinguishability property of ciphertexts of different messages, even when the plaintexts are chosen by the adversary. Therefore, semantic security against chosen plaintext attacks is referred to as IND-CPA security.

Definition 2.35 (IND-CPA [Vau06] (Section 9.3.7))

Semantic security against chosen plaintext attacks *(IND-CPA) in the context of public key cryptosystems is defined through the following game for a PPT adversary. The challenger generates a fresh key pair and gives the public key to the adversary, enabling him to perform encryption on any messages of his choice. The adversary selects two different plaintexts* (m_1, m_2) *and sends them to the challenger. The challenger randomly chooses one of the messages, encrypts it, and sends the resulting ciphertext to the adversary. Then, the adversary attempts to guess which message from* (m_1, m_2) *has been encrypted.*

A public key cryptosystem is IND-CPA if in the previous game, adversaries have negligible probability (in the security parameter κ*) to distinguish between an encryption of* m_1 *and an encryption of* m_2.

2.4.4 BBS Cryptosystem

Boneh, Boyen, and Shacham introduced in [BBS04] a cryptosystem based on the DLIN problem (see Section 2.2.2), which they called the *linear encryption*. In this thesis, it will be referred to as the BBS cryptosystem. Their scheme assumes the presence of a description of a bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ with a generator g_1 of $\mathbb{G}_1$. The secret key is composed by the pair $\mathsf{sk} = (\mathsf{sk}_1, \mathsf{sk}_2)$ such that $\mathsf{sk}_1, \mathsf{sk}_2 \in_R \mathbb{Z}_p^*$. The corresponding public key is the triplet $\mathsf{pk} = (g_1, f, h) = \left(g_1, g_1^{1/\mathsf{sk}_1}, g_1^{1/\mathsf{sk}_2}\right)$. To encrypt a message $m \in \mathbb{G}_1$ with randomness $r_f, r_h \in \mathbb{Z}_p^*$, the encryption algorithm outputs the ciphertext $c = (c_1, c_2, c_3) = \left(m \cdot g_1^{r_f + r_h},\ f^{r_f},\ h^{r_h}\right)$. To decrypt a ciphertext $c = (c_1, c_2, c_3)$, the decryption algorithm outputs $c_1 \left(c_2^{\mathsf{sk}_1} \cdot c_3^{\mathsf{sk}_2}\right)^{-1}$. Assuming the DLIN assumption holds, the BBS cryptosystem is semantically secure against a chosen plaintext attack (IND-CPA secure).

Lipmaa and Zhang proposed in [LZ12] a *lifted* version of the BBS cryptosystem. The modification brought by this lifted version, is to move the message m in the exponent of c_1, thus the

encryption becomes $c = (c_1, c_2, c_3) = \left(g_1^{m+r_f+r_h},\ f^{r_f},\ h^{r_h}\right)$. To decrypt, the discrete logarithm of $c_1\left(c_2^{\mathsf{sk}_1} \cdot c_3^{\mathsf{sk}_2}\right)^{-1}$ is returned. Therefore, decryption for large messages m is infeasible due to the DLog assumption. For the decryption to succeed, it is thus necessary for m to be small. Identically to the BBS cryptosystem, this lifted version is IND-CPA secure under the DLIN assumption.

2.4.5 Digital Signature Schemes

A digital signature scheme, sometimes formally called public-key signature scheme, is a scheme used to convince any verifier that a given user has "seen and approved" a given message. The concept of seeing and approving a message is sometimes referred to as *validating* a message. Moreover, the result of this validation process should be universally verifiable. More details on digital signature schemes can be found in Section 10.1 of [Vau06] and in Chapter 6 of [Gol04].

Definition 2.36 (Digital Signature Scheme)
A digital signature scheme *is the collection of three PPT algorithm* $(\text{Gen}, \text{Sign}, \text{Verif})$*, respectively the* key generator*, the* signature *and the* verification *algorithms. The key generator* $\text{Gen}(1^\kappa)$ *outputs* $(\mathsf{sk}, \mathsf{pk})$ *that correspond respectively to the secret and private key of the signature scheme. Let* $\mathbb{M}$ *be the message space of the signature scheme. The signature algorithm* $\text{Sign}(\mathsf{sk}, m)$ *takes as input the secret key of the signature scheme, a message* $m \in \mathbb{M}$ *to be signed, and outputs the signature* σ*. The verification algorithm* $\text{Verif}(\mathsf{pk}, m, \sigma)$ *takes as input the public key of the signature scheme, a message* m*, and a signature* σ *to be verified. If the verification of the signature* σ *is correct, then* Verif *outputs* 1*, otherwise it outputs* 0*. Where the verification of the signature is correct, the signature is said to be valid. A digital signature scheme is said to be correct if for every pair* $(\mathsf{pk}, \mathsf{sk})$ *generated by* Gen*,*

$$\Pr\left[(\mathsf{pk}, \mathsf{sk}) \leftarrow \text{Gen}\left(1^\kappa\right),\ \text{Verif}\left(\mathsf{pk},\ m,\ \text{Sign}(\mathsf{sk},\ m)\right) = 1\right] = 1.$$

The security of signature schemes is based on two properties: *unforgeability*, and *non-repudiation*. The unforgeability of signatures is the security property that protects against *forging* (finding or computing) a valid signature on a message $m \in \mathbb{M}$ without the secret key (see Section 6.1.4 of [Gol04]). This latter type of attack is called a *signature forgery* and is divided into several classes depending on the inputs and advantages of the attacker [Vau06]. Non-repudiation prevents signers from denying their valid signatures. As there is no formal definition for non-repudiation, and several interpretations for this notion co-exist, the following informal definition is given as a general understanding of it.

Definition 2.37 (Unforgeability)
A digital signature scheme $(\text{Gen}, \text{Sign}, \text{Verif})$ *is said to be* (existentially) unforgeable *if the following holds: any PPT adversary given* pk *will succeed to output a valid signature for a message* m *that has not been signed previously, with at most a negligible probability in* κ*.*

Definition 2.38 (Non-repudiation)
Given a valid signature σ on m that verifies with the public key pk *corresponding to* sk*, a* non-repudiable *digital signature scheme forbids the signer from denying having signed m, if he is the sole holder of* sk.

A naïve example of a digital signature scheme is the plain RSA signature scheme which is derived from the RSA public key cryptosystem [RSA78] (see Section 2.2.2). In the plain RSA signature scheme, a signature on message m is computed as $\sigma = m^d \pmod{n}$. The verification of σ is achieved by checking that $m \stackrel{?}{=} \sigma^e \pmod{n}$. Recall that the RSA modulus $n = pq$ is the product of two primes, and $1 = ed \pmod{(p-1)(q-1)}$ where (n,e) is the public key and (n,d) is the secret key. As explained in [Vau06] (Section 10.2.2), the plain RSA signature scheme suffers several security flaws (such as existential forgery) and needs to be modified in order to be used as a secure digital signature scheme.

Full Domain Hash (FDH) Signature Scheme

Bellare and Rogaway introduced in [BR93, BR96] the *full domain hash* (FDH) signature scheme. Its security proof was presented in the random oracle model and with the computational hardness assumption of the RSA problem (see Section 2.2.2 for an explanation of RSA and its related problem). The key generator $\mathrm{Gen}(1^\kappa)$ runs the key generator of the RSA public key cryptosystem and retrieves (n,e,d). It then outputs $(\mathsf{sk},\mathsf{pk})$ such that $\mathsf{sk} = (n,d)$ and $\mathsf{pk} = (n,e)$. Furthermore, it provides all parties an oracle access to a collision resistant hash function $\mathscr{F}$ that takes any string as input, and outputs for each new query a uniformly random element in $\mathbb{Z}_n^*$. Identical queries will result in the same output. The signature σ on a message m is obtained by computing $\sigma = (\mathscr{F}(m))^d \pmod{n}$. The verification process is achieved by checking if $\mathscr{F}(m) \stackrel{?}{=} \sigma^e \pmod{n}$. Lastly, note that Coron provided in [Cor00] a stronger security proof for the FDH signature scheme which enables the use of smaller RSA modulus.

Boneh-Boyen Signature Scheme

Boneh and Boyen introduced a signature scheme in [BB04] that is *existentially unforgeable* under a *weak chosen message attack*. Recall that existential forgery means that the adversary is able to produce a valid pair (m,σ) from the public key of the signature scheme, but without any control over which m will be produced.

Definition 2.39 (Weak Chosen Message Attack)
A weak chosen message attack is defined through the following game. The adversary begins by choosing and outputting q messages $(m_1,\ldots,m_q)$, before seeing the public key (hence the "weak" version)[6]*. The challenger generates a fresh key pair and gives the public key to the*

[6] In contrast, for a *chosen message attack* the selection of messages is done after seeing the public key.

adversary, together with signatures $(\sigma_1, \ldots, \sigma_q)$ *on* $(m_1, \ldots, m_q)$*. Then, the adversary attempts to output a valid signature* σ *on a message* $m \notin \{m_1, \ldots, m_q\}$*. If the adversary succeeds in doing so, it is said that he* wins the game.

Therefore, existential unforgeability under a weak chosen message attack means that no PPT adversary $\mathscr{A}$ has non-negligible probability of winning the game of the weak chosen message attack. Furthermore, the security of the Boneh-Boyen signature scheme is based on the q-Strong DH problem (see Section 2.2.2).

The Boneh-Boyen signature scheme is achieved as follows. The key generator $\text{Gen}(1^\kappa)$ outputs $\mathsf{sk} \in_R \mathbb{Z}_p^*$ and $\mathsf{pk} = (p, g, \mathbb{G}_1, \mathbb{G}_T, e, y)$, where $(p, g, \mathbb{G}_1, \mathbb{G}_T, e)$ are the parameters of a bilinear pairing, and $y = g^{\mathsf{sk}}$. The signature on a message m is achieved by computing $\sigma = g^{1/(\mathsf{sk}+m)}$. The verification is thus done by checking if $e(\sigma, y \cdot g^m) \stackrel{?}{=} e(g, g)$.

2.4.6 Cryptographic Accumulators

Benaloh and de Mare introduced the concept of cryptographic accumulators in [BdM93], which are algorithms that concentrate a large set of values into a single small element called the *accumulator*, such that for each value included in the accumulator there exists a corresponding witness to efficiently prove its membership, and such that it is infeasible to find such a witness for elements that were not included. Research in the field of accumulators has notably led to the results of Camenisch and Lysyanskaya in [CL02a], with *dynamic accumulators*, where elements can be added and removed at unit cost (independently of the number of accumulated elements) even after setting up an accumulator. The following accumulator definition is derived from [CL02a].

Definition 2.40 (Cryptographic Accumulator [CL02a])
A secure accumulator for a family of inputs $\{X_\kappa\}$ *is a family of families of functions* $\mathscr{G} = \{F_\kappa\}$ *with the following properties:*

Efficient generation: *There is a PPT algorithm* Gen *that on input* 1^κ *produces a random element* f *of* F_κ*, together with some auxiliary information about* f*, denoted* t_f*.*

Efficient evaluation: $f \in F_\kappa$ *is a polynomial-size circuit*[7] $f : U_f \times X_\kappa \to U_f$ *that, on input* $(u, x) \in U_f \times X_\kappa$*, outputs a value* $v \in U_f$*, where* U_f *is an efficiently samplable input domain for the function* f*; and* X_κ *is the intended input domain whose elements are to be accumulated.*

Quasi-commutative: *For all* κ*, for all* $f \in F_\kappa$*, for all* $u \in U_f$*, for all* $x_1, x_2 \in X_\kappa$*,* $f(f(u, x_1), x_2) = f(f(u, x_2), x_1)$*. If* $X = \{x_1, \ldots, x_m\} \subset X_\kappa$*, then* $f(u, X)$ *denotes* $f(f(\ldots(u, x_1), \ldots), x_m)$*.*

[7] Recall that a *circuit* is a computational model that processes inputs through a sequence of functions. See [Vol99] for a formal definition and more details. Although polynomial-size circuits could be simulated by an ITM with an advice string (see [AB09] and the complexity class $P/poly$), the definitions of cryptographic accumulators have historically used circuits as they are more convenient when handling functions with fixed length inputs.

Witnesses: *Let $v \in U_f$ and $x \in X_\kappa$. A value $w \in U_f$ is called a* witness *for x in v under f if $v = f(w,x)$.*

Security: *Let $U'_f \times X'_\kappa$ denote the domains for which the computational procedure for function $f \in F_\kappa$ is defined (thus $U_f \subseteq U'_f$, $X_\kappa \subseteq X'_\kappa$). For all PPT adversaries $\mathcal{A}_\kappa$,*

$$\Pr\left[\begin{array}{l} f \leftarrow \mathrm{Gen}(1^\kappa),\ u \in_R U_f,\ (x,w,X) \leftarrow \mathcal{A}_\kappa(f,U_f,u) : \\ X \subset X_\kappa,\ w \in U'_f,\ x \in X'_\kappa,\ x \notin X,\ f(w,x) = f(u,X) \end{array}\right]$$

is negligible in κ. Moreover, note that only the legitimate accumulated values $(x_1,\ldots,x_m)$ must belong to X_κ. The forged value x can belong to a possibly larger set X'_κ.

The accumulator of Camenisch and Lysyanskaya in [CL02a] recalled hereafter, is based on the strong RSA assumption (see Section 2.2.2). The main idea is to use modular exponentiation to incorporate prime elements into the accumulator, which is initially picked as a quadratic residue. Let F_κ be the family of functions that correspond to exponentiations modulo a special RSA modulus n of length κ. Recall that a special RSA modulus is the product of two safe primes. Choosing $f \in F_\kappa$ amounts to choosing a random modulus $n = pq$ of length κ, where $p = 2p'+1$, $q = 2q'+1$, and p,p',q,q' are all prime. Thus, the auxiliary information t_f is the factorization of n. The input domain for f is set to $U_f = \{u \in QR_n : u \neq 1\}$ and $U'_f = \mathbb{Z}_n^*$. Let X_κ be the set of primes $\{e : e \neq p', q' \ \wedge\ A \leqslant e \leqslant B\}$, where A and B can be chosen with arbitrary polynomial dependence on the security parameter κ, as long as $2 < A$ and $B < A^2$. Let X_κ be denoted as $X_{A,B}$ and let $X'_\kappa = X'_{A,B}$ be any subset of the set of integer from $[2, A^2-1]$ such that $X_{A,B} \subseteq X'_{A,B}$. The accumulating function f is thus

$$f(u,x) = u^x \pmod{n}$$

and is denoted by $f_{n,A,B}$ (or by f_n or f when it does not cause confusion) as the function f corresponding to modulus n and domain $X_{A,B}$. Lastly, note that

$$f(f(u,x_1),x_2) = f(f(u,x_2),x_1) = f(u,\{x_1,x_2\}) = u^{x_1 x_2} \pmod{n}.$$

2.4.7 Hadamard Product Argument

Assume that $(\mathrm{Gen}_{com}, \mathrm{Com})$ is a knowledge commitment scheme. Recall that a Hadamard product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ (of length n) is equal to their entrywise product vector $\boldsymbol{c}$, that is, $c_j = a_j \cdot b_j$ for $j \in \mathbb{Z}_n$. In a *Hadamard product argument*, the prover aims to convince the verifier that for three given commitments $(A,\widehat{A})$, $(B,\widehat{B})$, and $(C,\widehat{C})$, he knows how to open them as $(A,\widehat{A}) = \mathrm{Com}^1(\widehat{\mathsf{crs}}_1;\boldsymbol{a};r_a)$, $(B,\widehat{B}) = \mathrm{Com}^1(\widehat{\mathsf{crs}}_1;\boldsymbol{b};r_b)$, and $(C,\widehat{C}) = \mathrm{Com}^1(\widehat{\mathsf{crs}}_1;\boldsymbol{c};r_c)$, such that $c_j = a_j \cdot b_j$ for $j \in \mathbb{Z}_n$. The corresponding Hadamard product statement will be denoted as $[\![(C,\widehat{C})]\!] = [\![(A,\widehat{A})]\!] \circ [\![(B,\widehat{B},B_2)]\!]$, where B_2 is the equivalent of B in $\mathbb{G}_2$: $B_2 \leftarrow g_2^{r_b} \cdot$

$\prod_{i=0}^{n-1} g_{2,\lambda_i}^{b_i}$. Groth [Gro10] proposed an efficient (weakly)[8] sound and non-interactive witness indistinguishable (NIWI) Hadamard product argument that was refined by Lipmaa [Lip12a], who used the theory of progression free sets and asymmetric pairings to optimize the argument of Groth. Additionally, this Hadamard product argument has been further optimized in [FLZ13] and in [Lip14a]. Protocol 2.2 has a full description of the Hadamard product argument of Lipmaa [Lip12a]. More details can be found in [Lip11], which is the full version of [Lip12a]. The main idea of Protocol 2.2 is to construct an argument $\pi^{\times} \leftarrow (\pi, \widehat{\pi})$ such that $e(g_1, \pi) = e(A, B_2)/e(C, D)$, where D is a fixed public element in $\mathbb{G}_2$: $D \leftarrow \prod_{i=0}^{n-1} g_{2,\lambda_i}$. Furthermore, to prove that the prover knows how to open the commitment of the argument, $\pi^{\times}$ should also satisfy $e(g_1, \widehat{\pi}) \stackrel{?}{=} e(\widehat{g}_1, \pi)$.

System parameters: Let $n = poly(\kappa)$ be the length of the vectors in the Hadamard product. Let $\Lambda = \{\lambda_i : i \in \mathbb{Z}_n\}$ be a progression free set of odd integers, such that $\lambda_{i+1} > \lambda_i > 0$. Let $\widehat{\Lambda} := \{0\} \cup \Lambda \cup 2\widehat{\ }\Lambda$.

CRS generation $\mathsf{Gen}_{\mathsf{crs}\times}(1^{\kappa})$**:**
Let $\mathsf{param}_{\mathsf{bp}} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_{\mathrm{T}}, e) \leftarrow \mathsf{PG}_a(1^{\kappa})$. Let $\widehat{\alpha}, x \leftarrow \mathbb{Z}_p$. Let $g_1 \leftarrow \mathbb{G}_1 \setminus \{1\}$ and $g_2 \leftarrow \mathbb{G}_2 \setminus \{1\}$. Denote $g_{t,\ell} \leftarrow g_t^{x^{\ell}}$ and $\widehat{g}_{t,\ell} \leftarrow g_t^{\widehat{\alpha} x^{\ell}}$ for $t \in \{1,2\}$ and $\ell \in \widehat{\Lambda}$. Let $D \leftarrow \prod_{i=0}^{n-1} g_{2,\lambda_i}$. The CRS is $\mathsf{crs} \leftarrow (\mathsf{param}_{\mathsf{bp}}; (g_{1,\ell}, \widehat{g}_{1,\ell})_{\ell \in \{0\} \cup \Lambda}, (g_{2,\ell}, \widehat{g}_{2,\ell})_{\ell \in \widehat{\Lambda}}, D)$. Let $\widehat{\mathsf{crs}}_1 \leftarrow (\mathsf{param}_{\mathsf{bp}}; (g_{1,\ell}, \widehat{g}_{1,\ell})_{\ell \in \{0\} \cup \Lambda})$.

Common inputs: $(A, \widehat{A}, B, \widehat{B}, B_2, C, \widehat{C})$,
where $(A, \widehat{A}) \leftarrow \mathrm{Com}^1(\widehat{\mathsf{crs}}_1; \boldsymbol{a}; r_a)$, $(B, \widehat{B}) \leftarrow \mathrm{Com}^1(\widehat{\mathsf{crs}}_1; \boldsymbol{b}; r_b)$,
$B_2 \leftarrow g_2^{r_b} \cdot \prod_{i=0}^{n-1} g_{2,\lambda_i}^{b_i}$, $(C, \widehat{C}) \leftarrow \mathrm{Com}^1(\widehat{\mathsf{crs}}_1; \boldsymbol{c}; r_c)$, such that $a_i b_i = c_i$ for $i \in \mathbb{Z}_n$.

Argument generated by the prover:
$\text{NIZK-PK}\{(\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, \boldsymbol{c}, r_c) : [\![(C, \widehat{C})]\!] = [\![(A, \widehat{A})]\!] \circ [\![(B, \widehat{B}, B_2)]\!]\}$
Let $I_1(\ell) := \{(i, j) : i, j \in \mathbb{Z}_n \wedge j \neq i \wedge \lambda_i + \lambda_j = \ell\}$. For $\ell \in 2\widehat{\ }\Lambda$, the prover sets $\mu_{\ell} \leftarrow \sum_{(i,j) \in I_1(\ell)} (a_i b_j - c_i)$. He sets $\pi \leftarrow g_2^{r_a r_b} \cdot \prod_{i=0}^{n-1} g_{2,\lambda_i}^{r_a b_i + r_b a_i - r_c} \cdot \prod_{\ell \in 2\widehat{\ }\Lambda} g_{2\ell}^{\mu_{\ell}}$, and $\widehat{\pi} \leftarrow \widehat{g}_2^{r_a r_b} \cdot \prod_{i=0}^{n-1} \widehat{g}_{2,\lambda_i}^{r_a b_i + r_b a_i - r_c} \cdot \prod_{\ell \in 2\widehat{\ }\Lambda} \widehat{g}_{2\ell}^{\mu_{\ell}}$. He sends $\pi^{\times} \leftarrow (\pi, \widehat{\pi}) \in \mathbb{G}_2^2$ to the verifier as the argument.

Verification $\left(\mathsf{crs}; (A, \widehat{A}, B, \widehat{B}, B_2, C, \widehat{C}), \pi^{\times}\right)$**:**
The verifier checks that $e(A, B_2)/e(C, D) \stackrel{?}{=} e(g_1, \pi)$ and $e(g_1, \widehat{\pi}) \stackrel{?}{=} e(\widehat{g}_1, \pi)$.

Protocol 2.2 – Hadamard product argument $[\![(C, \widehat{C})]\!] = [\![(A, \widehat{A})]\!] \circ [\![(B, \widehat{B}, B_2)]\!]$ from [Lip12a]

Theorem 2.3 ([Lip12a](Theorem 4, Section 5))
The Hadamard product argument in Protocol 2.2 is perfectly complete and perfectly witness in-

[8] Note that here, the soundness is expressed in the inability for a PPT adversary to output an accepting argument together with openings to its corresponding commitments such that the restrictions for the Hadamard product argument are violated. Therefore this notion of soundness is weaker than computational soundness, where a PPT adversary is unable to provide an accepting argument from a false statement.

distinguishable. If the asymmetric bilinear group generator PG_a *is* $\widehat{\Lambda}$*-PSDL secure, then a non-uniform PPT adversary has negligible chance of outputting* $inp^{\times} \leftarrow (A, \widehat{A}, B, \widehat{B}, B_2, C, \widehat{C})$ *and an accepting argument* $\pi^{\times} \leftarrow (\pi, \widehat{\pi})$ *together with opening witness* $w^{\times} \leftarrow \left(\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, \boldsymbol{c}, r_c, (f'_s)_{s\in\widehat{\Lambda}}\right)$ *such that*

- $(A, \widehat{A}) = \mathrm{Com}^1(\widehat{\mathsf{crs}}_1; \boldsymbol{a}; r_a)$,
- $(B, \widehat{B}) = \mathrm{Com}^1(\widehat{\mathsf{crs}}_1; \boldsymbol{b}; r_b)$,
- $B_2 = g_2^{r_b} \cdot \prod_{i=0}^{n-1} g_{2i}^{b_i}$,
- $(C, \widehat{C}) = \mathrm{Com}^1(\widehat{\mathsf{crs}}_1; \boldsymbol{c}; r_c)$,
- $(\pi, \widehat{\pi}) = (g_2^{\sum_{s\in\widehat{\Lambda}} f'_s x^s}, \widehat{g}_2^{\sum_{s\in\widehat{\Lambda}} f'_s x^s})$,
- *and for some* $i \in \mathbb{Z}_n$, $a_i b_i \neq c_i$.

For the product argument to be useful in more complex arguments, the verifier should additionally check the validity of commitments: $e(A, \widehat{g}_2) = e(\widehat{A}, g_2)$, $e(B, \widehat{g}_2) = e(\widehat{B}, g_2)$, $e(g_1, B_2) = e(B, g_2)$, and $e(C, \widehat{g}_2) = e(\widehat{C}, g_2)$. Note that $(f'_s)_{s\in\widehat{\Lambda}}$ is the opening of $(\pi, \widehat{\pi})$. This can be seen as $(\pi, \widehat{\pi}) = (\prod_{s\in\widehat{\Lambda}} g_{2,s}^{f'_s}, \prod_{s\in\widehat{\Lambda}} \widehat{g}_{2,s}^{f'_s})$ and both $g_{2,s}$ and $\widehat{g}_{2,s}$ are in the common reference string crs.

Theorem 2.4 ([Lip12a](Theorem 5, Section 5))
For any $n > 0$ *and* $y = n^{1+o(1)}$*, let* $\Lambda \subset \mathbb{Z}_y$ *be a progression free set of odd integers as guaranteed by Theorem 2.1, such that* $|\Lambda| = n$*. The communication (argument size) of the Hadamard product argument is* 2 *elements from* $\mathbb{G}_2$*. The computational complexity of the prover is* $\Theta(n^2)$ *scalar multiplications in* $\mathbb{Z}_p$ *and* $n^{1+o(1)}$ *exponentiations in* $\mathbb{G}_2$*. The computational complexity of the verifier is dominated by* 5 *bilinear pairings. The CRS consists of* $n^{1+o(1)}$ *group elements.*

Finally, as noted in [Lip12a], if $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ are boolean vectors then the computational complexity of the prover is $\Theta(n^2)$ scalar additions in $\mathbb{Z}_p$ and $n^{1+o(1)}$ exponentiations in $\mathbb{G}_2$.

2.4.8 Lipmaa Permutation Argument

In a *permutation argument*, the prover aims to convince the verifier that for a given permutation ϱ from $\mathbb{Z}_n$ to $\mathbb{Z}_n$, and two commitments $(A, \widetilde{A})$ and $(B, \widetilde{B})$, he knows how to open them as $(A, \widetilde{A}) = \mathrm{Com}^1(\widetilde{\mathsf{crs}}_1; \boldsymbol{a}; r_a)$ and $(B, \widetilde{B}) = \mathrm{Com}^1(\widetilde{\mathsf{crs}}_1; \boldsymbol{b}; r_b)$, such that $b_j = a_{\varrho(j)}$ for $j \in \mathbb{Z}_n$. Denote this non-interactive argument by $\varrho\left(\left[\!\left[(A, \widetilde{A})\right]\!\right]\right) = \left[\!\left[(B, \widehat{B}, \widetilde{B})\right]\!\right]$, where the commitment $(B, \widehat{B})$ is equivalent to the commitment $(B, \widetilde{B})$ with respect to the CRS $\widehat{\mathsf{crs}}_1$: $(B, \widehat{B}) = \mathrm{Com}^1(\widehat{\mathsf{crs}}_1; \boldsymbol{b}; r_b)$. Groth [Gro10] proposed an efficient (weakly)[9] sound and non-interactive witness indistinguishable (NIWI) permutation argument that was further refined by Lipmaa [Lip12a], who

[9] Note that as for the Hadamard product argument, weakly soundness is defined here as the inability for a PPT adversary to output an accepting argument together with openings to its corresponding commitments such that the initial restrictions are violated.

used the theory of progression free sets and asymmetric pairings to optimize the argument of Groth. This permutation argument has been further improved by Fauzi et al. in [FLZ13] and the latest improvement has been provided by Lipmaa in [Lip14a]. The Lipmaa permutation argument in [Lip12a] is described in Protocol 2.3. Further details can be found in [Lip11], which is the full version of [Lip12a].

Let $T_\Lambda(i,\varrho) := \left|\left\{j \in \mathbb{Z}_n : 2\lambda_{\varrho(i)} + \lambda_j = 2\lambda_{\varrho(j)} + \lambda_i\right\}\right|$, clearly $T_\Lambda(i,\varrho) \geqslant 1$. The main idea of the product argument is to prove that $a_{\varrho(i)} = b_i$ for $i \in \mathbb{Z}_n$ by using two subarguments. The first one shows that for separately committed a_i^*, $a_i^* = T_\Lambda(\varrho^{-1}(i),\varrho)\cdot a_i$ for $i \in \mathbb{Z}_n$, which is equivalent to $a^*_{\varrho(i)} = T_\Lambda(i,\varrho)\cdot a_{\varrho(i)}$. This is achieved with a Hadamard product argument $(\pi^\times, \widehat{\pi}^\times)$ for $\left[\!\left[(A^*,\widehat{A}^*)\right]\!\right] = \left[\!\left[(A,\widehat{A})\right]\!\right] \circ \left[\!\left[(T^*,\widehat{T}^*,T_2^*)\right]\!\right]$. The second subargument shows that $a^*_{\varrho(i)} = T_\Lambda(i,\varrho)\cdot b_i$ for $i \in \mathbb{Z}_n$ and is achieved with the following verification $e(A^*,D)/e(B,E_\varrho) \stackrel{?}{=} e(g_1,\pi^\varrho)$ (see section 6 of [Lip11] for the analysis of completeness). Thus, from $a^*_{\varrho(i)} = T_\Lambda(i,\varrho)\cdot a_{\varrho(i)}$ and $a^*_{\varrho(i)} = T_\Lambda(i,\varrho)\cdot b_i$, one obtains that $a_{\varrho(i)} = b_i$ for $i \in \mathbb{Z}_n$. This permutation argument will be used only with fixed permutations ϱ and thus the element E_ϱ (and its counterpart $\widetilde{E}_\varrho$ in base $\widetilde{g}_2$) can be put in the CRS. Furthermore, Fauzi et al. noticed in [FLZ13] that the elements $\left(T^*,\widehat{T}^*,T_2^*\right)$ can also be put in the CRS, as they will be fixed by the permutation ϱ. Last but not least, notice that $\widehat{\Lambda}\cup\widetilde{\Lambda} = \{0\}\cup\widetilde{\Lambda}$, where $\widetilde{\Lambda}$ is defined in Protocol 2.3.

Theorem 2.5 ([Lip12a](Theorem 6, Section 6))

The permutation argument described in Protocol 2.3 is perfectly complete and perfectly witness indistinguishable. If the asymmetric bilinear group generator PG_a *is* $\widetilde{\Lambda}$*-PSDL secure, then a non-uniform PPT adversary has negligible chance of outputting* $inp^{perm} \leftarrow (A,\widetilde{A},B,\widehat{B},\widetilde{B},\varrho)$ *and an accepting argument* $\pi^{perm} \leftarrow (A^*,\widehat{A}^*,\pi^\times,\widehat{\pi}^\times,\pi^\varrho,\widetilde{\pi}^\varrho)$ *together with a witness*

$$w^{perm} \leftarrow (\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, \boldsymbol{a}^*, r_{a^*}, (f'_{(\times,\ell)})_{\ell\in\widehat{\Lambda}}, (f'_{(\varrho,\ell)})_{\ell\in\widetilde{\Lambda}}),$$

such that

- $\left(A,\widetilde{A}\right) = \mathrm{Com}^1\left(\widetilde{\mathsf{crs}}_1;\boldsymbol{a};r_a\right)$,
- $\left(B,\widehat{B}\right) = \mathrm{Com}^1\left(\widehat{\mathsf{crs}}_1;\boldsymbol{b};r_b\right)$,
- $\left(B,\widetilde{B}\right) = \mathrm{Com}^1\left(\widetilde{\mathsf{crs}}_1;\boldsymbol{b};r_b\right)$,
- $\left(A^*,\widehat{A}^*\right) = \mathrm{Com}^1\left(\widehat{\mathsf{crs}}_1;\boldsymbol{a}^*;r_{a^*}\right)$,
- $(\pi^\times,\widehat{\pi}^\times) = \left(g_2^{\sum_{\ell\in\widehat{\Lambda}} f'_{(\times,\ell)}}, \widehat{g}_2^{\sum_{\ell\in\widehat{\Lambda}} f'_{(\times,\ell)}}\right)$,
- $(\pi^\varrho,\widehat{\pi}^\varrho) = \left(g_2^{\sum_{\ell\in\widetilde{\Lambda}} f'_{(\varrho,\ell)}}, \widetilde{g}_2^{\sum_{\ell\in\widetilde{\Lambda}} f'_{(\varrho,\ell)}}\right)$,
- $a_i^* = T_\Lambda\left(\varrho^{-1}(i),\varrho\right)\cdot a_i$, *for* $i \in \mathbb{Z}_n$, *and*
- *for some* $i \in \mathbb{Z}_n$, $a_{\varrho(i)} \neq b_i$.

For the permutation argument to be useful in more complex arguments, the verifier should additionally check the validity of commitments: $e(\widetilde{A},g_2) = e(A,\widetilde{g}_2)$, $e(\widehat{B},g_2) = e(B,\widehat{g}_2)$, and $e(\widetilde{B},g_2) = e(B,\widetilde{g}_2)$.

System parameters: Same as in Protocol 2.2, but let
$\widetilde{\Lambda} := \Lambda \cup \{2\lambda_k - \lambda_j\}_{i,k\in\mathbb{Z}_n} \cup 2\widehat{\ }\Lambda \cup (\{2\lambda_k + \lambda_i - \lambda_j\}_{i,j,k\in\mathbb{Z}_n \wedge i\neq j} \setminus 2\cdot\Lambda)$.

CRS generation $\mathrm{Gen}_{\mathsf{crs}_perm}(1^\kappa)$**:**
Let $\mathsf{param}_{\mathsf{bp}} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathsf{PG}_a(1^\kappa)$. Let $\widehat{\alpha}, \widetilde{\alpha}, x \leftarrow \mathbb{Z}_p$. Let $g_1 \leftarrow \mathbb{G}_1 \setminus \{1\}$ and $g_2 \leftarrow \mathbb{G}_2 \setminus \{1\}$. Let $\widehat{g}_t \leftarrow g_t^{\widehat{\alpha}}$ and $\widetilde{g}_t \leftarrow g_t^{\widetilde{\alpha}}$ for $t \in \{1,2\}$.
Denote $g_{t,\ell} \leftarrow g_t^{x^\ell}$, $\widehat{g}_{t,\ell} \leftarrow \widehat{g}_t^{x^\ell}$, and $\widetilde{g}_{t,\ell} \leftarrow \widetilde{g}_t^{x^\ell}$ for $t \in \{1,2\}$ and $\ell \in \{0\} \cup \widetilde{\Lambda}$.
Let $(D, \widetilde{D}) \leftarrow \left(\prod_{i=0}^{n-1} g_{2,\lambda_i}, \prod_{i=0}^{n-1} \widetilde{g}_{2,\lambda_i}\right)$.
The CRS is
$\mathsf{crs} \leftarrow (\mathsf{param}_{\mathsf{bp}}; (g_{1,\ell}, \widehat{g}_{1,\ell}, \widetilde{g}_{1,\ell})_{\ell\in\{0\}\cup\Lambda}, (g_{2,\ell})_{\ell\in\{0\}\cup\widetilde{\Lambda}}, (\widehat{g}_{2,\ell})_{\ell\in\widehat{\Lambda}}, (\widetilde{g}_{2,\ell})_{\ell\in\widetilde{\Lambda}}, D, \widetilde{D})$.
Let $\widehat{\mathsf{crs}}_1 \leftarrow \left(\mathsf{param}_{\mathsf{bp}}; (g_{1,\ell}, \widehat{g}_{1,\ell})_{\ell\in\{0\}\cup\Lambda}\right)$, $\widetilde{\mathsf{crs}}_1 \leftarrow \left(\mathsf{param}_{\mathsf{bp}}; (g_{1,\ell}, \widetilde{g}_{1,\ell})_{\ell\in\{0\}\cup\Lambda}\right)$.

Common inputs: $(A, \widetilde{A}, B, \widehat{B}, \widetilde{B}, \varrho)$,
where ϱ is a permutation from $\mathbb{Z}_n$ to $\mathbb{Z}_n$, $(A, \widetilde{A}) \leftarrow \mathrm{Com}^1(\widetilde{\mathsf{crs}}_1; \boldsymbol{a}; r_a)$,
$(B, \widehat{B}) \leftarrow \mathrm{Com}^1(\widehat{\mathsf{crs}}_1; \boldsymbol{b}; r_b)$, and $(B, \widetilde{B}) \leftarrow \mathrm{Com}^1(\widetilde{\mathsf{crs}}_1; \boldsymbol{b}; r_b)$,
such that $b_j = a_{\varrho(j)}$ for $j \in \mathbb{Z}_n$.

Argument generated by the prover:
$\text{NIZK-PK}\{(\boldsymbol{a}, r_a, \boldsymbol{b}, r_b) : \varrho\left(\left[\!\left[(A, \widetilde{A})\right]\!\right]\right) = \left[\!\left[(B, \widehat{B}, \widetilde{B})\right]\!\right]\}$

1. Let $(T^*, \widehat{T}^*, T_2^*) \leftarrow (\prod_{i=0}^{n-1} g_{1,\lambda_i}^{T_\Lambda(\varrho^{-1}(i),\varrho)}, \prod_{i=0}^{n-1} \widehat{g}_{1,\lambda_i}^{T_\Lambda(\varrho^{-1}(i),\varrho)}, \prod_{i=0}^{n-1} g_{2,\lambda_i}^{T_\Lambda(\varrho^{-1}(i),\varrho)})$.
2. Let $r_{a^*} \leftarrow \mathbb{Z}_p$,
$(A^*, \widehat{A}^*) \leftarrow \mathrm{Com}^1(\widehat{\mathsf{crs}}_1; T_\Lambda(\varrho^{-1}(0),\varrho)\cdot a_0, \ldots, T_\Lambda(\varrho^{-1}(n-1),\varrho)\cdot a_{n-1}; r_{a^*})$.
Create an argument $(\pi^\times, \widehat{\pi}^\times)$ for $\left[\!\left[(A^*, \widehat{A}^*)\right]\!\right] = \left[\!\left[(A, \widehat{A})\right]\!\right] \circ \left[\!\left[(T^*, \widehat{T}^*, T_2^*)\right]\!\right]$.
3. Let $\widetilde{\Lambda}'_\varrho := 2\widehat{\ }\Lambda \cup (\{2\lambda_{\varrho(j)} + \lambda_i - \lambda_j : i, j \in \mathbb{Z}_n \wedge i \neq j\} \setminus 2\cdot\Lambda) \subset \{-\lambda_{n-1}+1, \ldots, 3\lambda_{n-1}\}$.
4. For $\ell \in \widetilde{\Lambda}'_\varrho$, set $I_1(\ell)$ as in Protocol 2.2,
$I_2(\ell) := \left\{(i,j) : \begin{array}{l} i, j \in \mathbb{Z}_n \ \wedge\ j \neq i \ \wedge\ 2\lambda_{\varrho(i)} + \lambda_j \neq \lambda_i + 2\lambda_{\varrho(j)} \\ \wedge\ 2\lambda_{\varrho(j)} + \lambda_i - \lambda_j = \ell \end{array}\right\}$,
and
$$\mu_{\varrho,\ell} \leftarrow \sum_{(i,j)\in I_1(\ell)} a_i^* - \sum_{(i,j)\in I_2(\ell)} b_i.$$
5. Let $(E_\varrho, \widetilde{E}_\varrho) \leftarrow (\prod_{i=0}^{n-1} g_{2,2\lambda_{\varrho(i)}-\lambda_i}, \prod_{i=0}^{n-1} \widetilde{g}_{2,2\lambda_{\varrho(i)}-\lambda_i})$.
6. Let $\pi^\varrho \leftarrow D^{r_a^*} \cdot E_\varrho^{-r_b} \cdot \prod_{\ell\in\widetilde{\Lambda}'_\varrho} g_{2,\ell}^{\mu_{\varrho,\ell}}$, $\widetilde{\pi}^\varrho \leftarrow \widetilde{D}^{r_a^*} \cdot \widetilde{E}_\varrho^{-r_b} \cdot \prod_{\ell\in\widetilde{\Lambda}'_\varrho} \widetilde{g}_{2,\ell}^{\mu_{\varrho,\ell}}$,

Send $\pi^{perm} \leftarrow (A^*, \widehat{A}^*, \pi^\times, \widehat{\pi}^\times, \pi^\varrho, \widetilde{\pi}^\varrho) \in \mathbb{G}_1^2 \times \mathbb{G}_2^4$ to the verifier as the argument.

Verification $\left(\mathsf{crs}; (A, \widetilde{A}, B, \widehat{B}, \widetilde{B}, \varrho), \pi^{perm}\right)$**:**
Let $(E_\varrho, \widetilde{E}_\varrho)$ and $(T^*, \widehat{T}^*, T_2^*)$ be computed as above. If $(\pi^\times, \widehat{\pi}^\times)$ verifies, $e(A^*, D)/e(B, E_\varrho) = e(g_1, \pi^\varrho)$, $e(A^*, \widehat{g}_2) = e(\widehat{A}^*, g_2)$, and $e(g_1, \widetilde{\pi}^\varrho) = e(\widetilde{g}_1, \pi^\varrho)$, then the verifier accepts the argument. Otherwise, the verifier rejects it.

Protocol 2.3 – Permutation argument $\varrho\left(\left[\!\left[(A, \widetilde{A})\right]\!\right]\right) = \left[\!\left[(B, \widehat{B}, \widetilde{B})\right]\!\right]$ from [Lip12a]

Theorem 2.6 ([Lip12a](Theorem 7, Section 6))
The permutation argument has a common reference string of length $n^{1+o(1)}$ *and communication of* 6 *group elements (*2 *elements from* $\mathbb{G}_1$ *and* 4 *elements from* $\mathbb{G}_2$*). The computational complexity of the prover is* $\Theta(n^2)$ *scalar additions in* $\mathbb{Z}_p$ *and* $n^{1+o(1)}$ *exponentiations in* $\mathbb{G}_2$*. The computational complexity of the verifier is dominated by* 12 *bilinear pairings.*

2.5 Threshold Cryptosystems

Threshold cryptosystems are a fundamental notion for the terminal revocation solution presented in Chapter 7. In the following, the focus is set on the case of threshold RSA signatures as they provide a good balance between security and efficiency considering the protocol participants in the Extended Access Control (see Chapter 6). Indeed, the computational power of Machine Readable Travel Documents (such as e-passports) is much more limited compared to the one of terminals. Furthermore, this threshold signature scheme is based on secret sharing, which is explained hereinafter.

2.5.1 Secret Sharing

The notion of secret sharing was introduced by Shamir in [Sha79] and independently by Blakley in [Bla79]. It aims at dividing the knowledge of a secret among ℓ servers. The motivation behind it was to protect a secret against the corruption of some servers. To achieve secret sharing, Shamir used Lagrange interpolation to divide the secret into multiple shares. The main idea is that any polynomial function f of degree t can be reconstructed from $t+1$ distinct points. $f(0)$ is considered to be the secret s to be shared. If given t or less points, the function cannot be reconstructed. Hence every participant will be given a point of the function as a secret share. Mathematically speaking, f is defined as follows:

$$f(x) = \sum_{i=0}^{t} a_i \cdot x^i. \tag{2.1}$$

As mentioned, the secret is $s = f(0) = a_0$. Every participant $i > 0$ will be provided the secret share $s_i = f(i)$. Given a set of participants Ψ with $|\Psi| = t+1$, f can be reconstructed as follows:

$$f(x) = \sum_{j \in \Psi} s_j \cdot \lambda_{x,j}^{\Psi}, \tag{2.2}$$

where $\lambda_{x,j}^{\Psi}$ are the Lagrange coefficients defined by

$$\lambda_{x,j}^{\Psi} = \prod_{i \in \Psi \setminus j} (i-x) \cdot (i-j)^{-1} \tag{2.3}$$

and with $x \notin \Psi$. Moreover, note that the inversion present in the Lagrange coefficients requires working in a field. This restriction can be relaxed although there would be a loss in efficiency. For instance, the computation of $f(x)$ could be replaced with the computation of $i_{max}! \cdot f(x)$, where i_{max} is larger or equal than the largest index among participants.

When secret sharing is used, the participant or authority in charge of reconstructing the secret will obviously learn the secret. This is of course not a desirable property and to be instantiated in practice, secret sharing needs some modifications. Ideally, the goal would be to share (to divide) the secret among ℓ servers, with the constraint that servers could perform computations based on the secret without reconstructing it. One such application is threshold cryptography, with the additional constraint that to be able to use the secret, $t+1$ servers need to collaborate. Thus, using the secret is achieved without being able to reconstruct it. Furthermore, no t or less servers could use the secret. The solution presented in Chapter 7 uses threshold signatures rather than threshold decryption. Nevertheless, the latter could still be used although negatively impacting efficiency.

2.5.2 Threshold Signatures

Participants in a threshold signature scheme consist of ℓ *signers* $\mathcal{S}_i$, a *trusted dealer* $\mathcal{T}$, a verifier $\mathcal{V}$, and an *adversary* $\mathcal{A}$. The scheme itself is composed of a set of five algorithms: $(KG, \Sigma_i, \Sigma_v, \Sigma_c, V_\sigma)$. The trusted dealer $\mathcal{T}$ runs KG in order to generate all the parameters and keys of the threshold signature scheme. Then $\mathcal{T}$ publishes the public parameters and sends to each $\mathcal{S}_i$ its respective secret key. Each signer $\mathcal{S}_i$ is thus able to create a partial signature by running algorithm Σ_i. The resulting partial signatures can be verified with algorithm Σ_v. To combine the necessary partial signatures into the general signature σ, a signer $\mathcal{S}_i$ runs algorithm Σ_c. Lastly, the signature σ can be verified with algorithm V_σ. In the following, a description of these five algorithms is provided.

Key generation: $KG(1^\kappa, t, \ell) \longrightarrow (\mathsf{pk}, \{\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell\}, \{\mathsf{vk}_1, \ldots, \mathsf{vk}_\ell\}, \mathsf{vk})$.

The *key generation algorithm*, run by $\mathcal{T}$, takes the security parameter κ, the threshold parameter t, and the number of participants ℓ as input. It outputs the public key pk of the system, ℓ secret keys sk_i together with their corresponding verification keys vk_i and the general verification key vk of the system.

Partial signing: $\Sigma_i(m, \mathsf{pk}, \mathsf{vk}, \mathsf{sk}_i, \mathsf{vk}_i) \longrightarrow (\sigma_i, [\pi_i])$.

The *partial signature algorithm*, run by $\mathcal{S}_i$, takes as input a message m, the general public key pk, the general verification key vk, and the secret share sk_i with its verification key vk_i. It outputs a partial signature σ_i with an optional verification proof π_i on the validity of σ_i. For reasons of simplicity, the notation $\Sigma_i(m, \mathsf{pk}, \mathsf{vk}, \mathsf{sk}_i, \mathsf{vk}_i)$ is shortened to $\Sigma_i(m)$ when there is no confusion.

Partial signature verification: $\Sigma_v(m, \mathsf{pk}, \mathsf{vk}, \sigma_i, \pi_i, \mathsf{vk}_i) \longrightarrow \{0, 1\}$.

The *partial signature verification algorithm*, run by any verifier $\mathcal{V}$, takes as input a mes-

sage m, the general public key pk, the general verification key vk, the partial signature σ_i, its corresponding verification proof π_i and verification key vk_i. It checks the validity of σ_i and outputs the result. The verification of σ_i with π_i is used to achieve robustness. For simplicity reasons, the notation $\Sigma_v(m, \mathsf{pk}, \mathsf{vk}, \sigma_i, \pi_i, \mathsf{vk}_i)$ is shortened to $\Sigma_v(\sigma_i, \pi_i)$ when there is no confusion.

Partial signature combining: $\Sigma_c(m, \mathsf{pk}, \{\sigma_i\}_\Psi) \longrightarrow \sigma$.

The *combining share algorithm*, run by any $\mathscr{S}_i$, takes as input a message m, the public key pk, and a set Ψ of size $t+1$ of valid partial signatures σ_i. It outputs the signature σ of m. For simplicity reasons, the notation $\Sigma_c(m, \mathsf{pk}, \{\sigma_i\}_\Psi)$ is shortened to $\Sigma_c(\{\sigma_i\}_\Psi)$ when there is no confusion.

Signature verification: $V_\sigma(m, \sigma, \mathsf{pk}) \longrightarrow \{0,1\}$.

The *signature verification algorithm*, run by any verifier $\mathscr{V}$, takes as input a message m, its signature σ, and the public key pk. It checks the validity of σ and outputs the result. For simplicity reasons, the notation $V_\sigma(m, \sigma, \mathsf{pk})$ is shortened to $V_\sigma(m)$ when there is no confusion.

Threshold Signature Security Requirements. The security requirements for threshold signatures are *robustness*, *threshold security*, *existential unforgeability*, and optionally *proactive security*.

- *Robustness* states that if all partial signatures used to create a signature σ on message m are valid then the signature σ is a valid signature of m.
- The *threshold security* requirement states that any authorized subsets of $t+1$ or more signers can produce a valid threshold signature on message m, but no other coalitions. This imply that no subset of t or less signers can produce a valid threshold signature.
- A threshold signature scheme is said to be *existentially unforgeable* if a computationally bounded adversary is unable to perform an adaptive chosen message attack. The goal of such an attack is to forge a valid partial signature or a valid signature on a chosen message, while having access to a signature oracle and while taking into account the responses from this signature oracle. Only in the case of forging a valid signature, the adversary is additionally allowed to corrupt up to t signers.
- *Proactive security* states that an update mechanism exists for signers to update their secret key share, without modifying the general public key of the system (nor the general verification key).

2.5.3 Threshold RSA

As the plain RSA signature scheme [RSA78] suffers several security flaws (see [Vau06], Section 10.2.2), threshold RSA is usually based on a variant of the plain RSA signature scheme, such as the Full Domain Hash (FDH) signature scheme from Bellare and Rogaway [BR93,

BR96, Cor00] (see Section 2.4.5). Let p and q be two large primes such that $n = pq$. Let $ed \pmod{\varphi(n)} = 1$, where φ is the Euler's Totient function. Hence $\varphi(n) = (p-1)(q-1)$. The public key of this system is (n, e). Let $\mathscr{F}$ be the hash function used in the FDH signature scheme. Hence $\mathscr{F}$ hashes from any message space $\mathscr{M}$ into the full domain $\mathbb{Z}_n^*$. To obtain a signature σ on a message m, the signer computes $\sigma = (\mathscr{F}(m))^d \pmod{n}$. Hence d is part of the private key of the signer. To verify the signature, it suffices to check the following: $\mathscr{F}(m) \stackrel{?}{=} \sigma^e \pmod{n}$.

To obtain the threshold version of the FDH variant of the RSA signature scheme, the secret d needs to be shared among ℓ servers. Assume the presence of a trusted party in charge of the key generation algorithm. In the case of the Extended Access Control (Chapter 7), this trusted party will be the Document Verifier (DV). To share d, secret sharing will be used. However this cannot be done directly as revealing $\varphi(n)$ to the signers would allow them to factorize n and thus compute d from e. Hence a single signer would be able to sign on behalf of the entire group. Extensive research has been undertaken regarding threshold RSA signatures and the solution regarding terminal revocation proposed in Chapter 7 is based on the threshold RSA signature of Shoup [Sho00]. Depending on the number of servers ℓ and the threshold value t, solutions from King [Kin00] and Desmedt-Frankel [DF94] could also be considered.

Shoup [Sho00] suggested the use of safe primes for the RSA modulus. Hence $n = pq = (2p'+1)(2q'+1)$ such that p, p', q and q' are primes. Let $\widetilde{n} = p'q'$. The value of $\widetilde{n}$ will not be revealed and should be kept secret from all parties. If proactive security is not needed, then $\widetilde{n}$ can be safely erased after the key generation phase. The public exponent e will be chosen as a prime with $e > \ell$. d will be picked such that $ed \equiv 1 \pmod{\widetilde{n}}$ and shared using the secret sharing of Shamir (see Section 2.5.1). Hence the secret share of signer i will be of the form $d_i = f(i) \pmod{\widetilde{n}}$, where f is defined by the equation (2.1) of Section 2.5.1. Let QR_n be the set of all quadratic residues modulo n. Recall that QR_n is thus the subgroup of squares in $\mathbb{Z}_n^*$. The general verification key vk will be randomly chosen in QR_n. The verification key of signer i will be set as $\mathsf{vk}_i = \mathsf{vk}^{d_i} \in QR_n$. To compute the Lagrange coefficients, the trick explained at the end of Section 2.5.1 is used with $i_{max} = \ell$. Let $\Delta = (\ell!)$.

To generate a partial signature σ_i on message m, signer i will first compute $x = \mathscr{F}(m)$ and then $\sigma_i = x^{2\Delta d_i}$. The validity proof π_i of σ_i consists of proving the statement $\log_{\mathsf{vk}}(\mathsf{vk}_i) = \log_{x^{4\Delta}}\left(\sigma_i^2\right)$. This can be achieved with a small variant of the NIZK proof of Chaum and Pedersen in [CP92], for proving discrete logarithm equality in the random oracle model. Note that the requirement for a variation originates from the fact that computations are performed in a group of unknown order.

Protocol 2.4 illustrates the discrete logarithm equality argument π_i, which is briefly described hereafter. Recall that the prover knows the discrete logarithm $\mathsf{sk}_i = d_i = \log_{\mathsf{vk}}(\mathsf{vk}_i)$, however the order of vk is unknown. The goal of the prover (the signer in the threshold RSA) is to convince verifiers that given the two group elements $(R, S) = \left(x^{4\Delta}, \sigma_i^2\right)$, the following statement holds: $\log_{\mathsf{vk}} \mathsf{vk}_i = \log_R S$. To do so, the prover picks a sufficiently large random $a \in_R \mathbb{Z}_{2^{\|n\|+2L_1}}$,

Common inputs: $(\mathscr{H}, \mathsf{vk}, \mathsf{vk}_i, R, S, L_1)$,
where L_1 is a security parameter, $\mathscr{H} : QR_n^6 \rightarrow 1^{L_1}$ is a hash function,
$\mathsf{vk}, \mathsf{vk}_i \in QR_n$ are verification keys such that $\mathsf{vk}_i = \mathsf{vk}^{\mathsf{sk}_i}$,
$R, S \in QR_n$ such that $R = x^{4\Delta}$ and $S = \sigma_i^2$.

Argument generated by the prover:
NIZK-PK$\left\{(\mathsf{sk}_i) : \mathsf{vk}_i = \mathsf{vk}^{\mathsf{sk}_i} \wedge \sigma_i^2 = (x)^{4\Delta \mathsf{sk}_i}\right\}$
The prover picks a large element $a \in_R \mathbb{Z}_{2^{\|n\|+2L_1}}$. Let $A = \mathsf{vk}^a$ and $B = R^a$. The prover computes $c = \mathscr{H}(\mathsf{vk}, R, \mathsf{vk}_i, S, A, B)$ and $z = a + c \cdot \mathsf{sk}_i$.
He sends $\pi_i \leftarrow (c, z)$ to the verifier as the argument.

Verification $(\mathscr{H}, \mathsf{vk}, \mathsf{vk}_i, R, S, \pi_i)$**:**
The verifier checks that $c \stackrel{?}{=} \mathscr{H}(\mathsf{vk},\ R,\ \mathsf{vk}_i,\ S,\ \mathsf{vk}^z \mathsf{vk}_i^{-c},\ R^z S^{-c})$.

Protocol 2.4 – Discrete logarithm equality argument $\log_{\mathsf{vk}}(\mathsf{vk}_i) = \log_{x^{4\Delta}}\left(\sigma_i^2\right)$
(variant of Chaum and Pedersen [CP92])

where L_1 is a secondary security parameter (Shoup suggests $L_1 = 128$)[10], and computes $A = \mathsf{vk}^a$, $B = R^a$, $c = \mathscr{H}(\mathsf{vk}, R, \mathsf{vk}_i, S, A, B)$, and $z = a + c \cdot \mathsf{sk}_i$. Note that $\mathscr{H}$ is a hash function that maps six group elements to an L_1 bit integer. The NIZK proof will thus consist of $\pi_i = (c, z)$. Indeed, any verifier can be convinced of the veracity of the statement by checking if $c \stackrel{?}{=} \mathscr{H}(\mathsf{vk}, R, \mathsf{vk}_i, S, \mathsf{vk}^z \mathsf{vk}_i^{-c}, R^z S^{-c})$. However, this proof would lose soundness if the prover is free to choose vk_i. More details can be found in Section 3 of [BPW12]. In the case of the solution presented in Chapter 7, vk_i is fixed by a trusted third party and then given to the prover.

Lastly, combining the $t+1$ valid partial signatures σ_j, with $j \in \Psi$ and $\Psi \subset \{1, \ldots, \ell\}$, means computing the signature $\sigma = \left(\prod_{j \in \Psi} \sigma_j^{2\Delta\lambda_{0,j}^{\Psi}}\right)^{\alpha} x^{\beta} \pmod{n}$, where α and β are obtained by solving $\alpha \cdot 4\Delta^2 + \beta \cdot e = 1 \pmod{\widetilde{n}}$ with the extended Euclidean algorithm. Notice that α and β can be precomputed by the trusted authority. Finally, the verification of σ is the same as in the FDH signature scheme.

Therefore, the algorithms composing the threshold RSA signature presented by Shoup [Sho00] are as follows:

- $KG(1^\kappa, t, \ell) \longrightarrow (\mathsf{pk}, \{\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell\}, \{\mathsf{vk}_1, \ldots, \mathsf{vk}_\ell\}, \mathsf{vk})$:
  $\mathsf{pk} = \left(n, e, \Delta, \mathscr{F}, \alpha, \beta\right)$, where $n, e, \mathscr{F}$ are obtained from the parameters of the FDH signature scheme, $\Delta = \ell!$, and α, β are obtained by solving $\alpha \cdot 4\Delta^2 + \beta \cdot e = 1 \pmod{\widetilde{n}}$ with the extended Euclidean algorithm.

[10] Note that according to Shoup, the coefficient 2 in front of L_1 is set in order to strengthen the zero-knowledge simulatability.

$\mathsf{sk}_i = d_i = \sum_{j=0}^{t} a_j \cdot i^j \pmod{\widetilde{n}}$, $\forall i \in \{1,\ldots,\ell\}$, with $a_0 = d$ and $a_{j>0} \in_R \mathbb{Z}_{\widetilde{n}}$.
$\mathsf{vk} \in_R QR_n$ and $\mathsf{vk}_i = \mathsf{vk}^{\mathsf{sk}_i}$, $\forall i \in \{1,\ldots,\ell\}$.

- $\Sigma_i(m, \mathsf{pk}, \mathsf{vk}, \mathsf{sk}_i, \mathsf{vk}_i) \longrightarrow (\sigma_i, [\pi_i])$:
  $\sigma_i = (\mathscr{F}(m))^{2\Delta \mathsf{sk}_i}$.
  $\pi_i = \text{NIZK-PK}\left\{(\mathsf{sk}_i) : \mathsf{vk}_i = \mathsf{vk}^{\mathsf{sk}_i} \wedge \sigma_i^2 = (\mathscr{F}(m))^{4\Delta \mathsf{sk}_i}\right\}$.
- $\Sigma_v(m, \mathsf{pk}, \mathsf{vk}, \sigma_i, \pi_i, \mathsf{vk}_i) \longrightarrow \{0,1\}$.
  The partial signature verification algorithm checks the NIZK-PK π_i.
- $\Sigma_c(m, \mathsf{pk}, \{\sigma_i\}_\Psi) \longrightarrow \sigma$.
  $\sigma = \left(\prod_{j \in \Psi} \sigma_j^{2\Delta\lambda_{0,j}^{\Psi}}\right)^{\alpha} (\mathscr{F}(m))^{\beta} \pmod{n}$, where $\lambda_{0,j}^{\Psi}$ are computed as in equation (2.3).
- $V_\sigma(m, \sigma, \mathsf{pk}) \longrightarrow \{0,1\}$.
  The signature verification algorithm performs the following check: $\mathscr{F}(m) \stackrel{?}{=} \sigma^e \pmod{n}$

More details as well as the security proof of the threshold RSA signature scheme of Shoup can be found in [Sho00].

Part I

Set Membership and Range Proofs

Chapter 3

Set Membership Proofs

In this chapter, we first present in Section 3.1 the set membership proof primitive. Section 3.2 introduces prior work achieved in this field, explains subsequent results, emphasizes the use of cryptographic accumulators, and presents some related work. Section 3.3 then gives a solution for the set membership proof based on the Boneh-Boyen signature scheme [BB04]. Section 3.4 shows that other signature schemes could also be used. This is demonstrated using the signature scheme proposed by Camenisch and Lysyanskaya in [CL02b]. Last but not least, Section 3.5 presents an alternative solution based on accumulators. The results of Section 3.3 and Section 3.5 are published at Asiacrypt 2008 [CCs08], as a joint work with Jan Camenisch and abhi shelat[1]. Even though the theoretical idea behind Section 3.4 was set out in [CCs08], the actual protocol is detailed exclusively here.

3.1 Set Membership Proof Primitive

The problem we are trying to solve in this chapter is called the *set membership proof* problem. It can be easily explained using the following game. Consider the existence of a public set Φ and two players. The public set could be for instance a set of names or a set of parameters. The first player secretly chooses an element σ from the public set, and digitally commits to it. Let us name the first player the *prover*. As his name indicates, his goal is to *prove* a specific statement to the second player, who becomes the *verifier*. The statement that the *prover* wants to *prove* is that the element he has picked (and that he has committed to) is indeed *contained* in the public set. In other words, he wants to prove that the element contained in his commitment, is a member of the public set. However, two main constraints are involved. On the one hand, the prover wants to keep his value secret, revealing no additional information besides the *set membership proof* of his committed element and the fact that he is able to open his commitment to such a value. However, on the other hand, the verifier wants to be sure that

[1] Note that abhi shelat requires his name to be written in lower case.

the prover is unable to cheat. The first constraint can be achieved by way of a *zero-knowledge* property, while the second constraint needs to be insured with a *soundness* property.

The main motivation behind *set membership* proofs as a cryptographic building block, comes from the challenges brought when cryptographic protocols in an idealized model need to be adapted to face malicious adversaries. Furthermore, they are also important for other applications, such as in the context of *anonymous credentials*. Consider a user who is issued a credential containing a number of attributes such as an address, and assume that the user needs to prove that she lives in a European capital. In this case, a list of all such cities is given and the user has to show that she possesses a credential containing one of those cities as an address (without of course, leaking the city the user lives in). Another example is where a user who has a subscription to a journal (for instance the news and the sports sections). Assume that some general sections are only accessible to subscribers of specific lists. Using a set membership proof, the user can efficiently show that she is a subscriber to one of the required kinds. *Online card games* also need set membership proofs, in order to prove that a given card, played face down, is a valid card without revealing the value of the card. This need has been mentioned by Barnett and Smart in [BS03]. Additionally, the need for set membership proofs also arises in an *electronic election* or an *e-voting* scheme, as pointed out by Cramer et al. in [CGS97]. Assume that in an electronic election or an e-voting scheme, a user is required to prove that his ballot contains a valid name or a valid vote respectively. These proofs are straightforward to solve by a set membership proof.

Definition 3.1 (Proof of Set Membership)
Let $C = (\text{Gen}, \text{Com}, \text{Open})$ *be the generation, the commit and the open algorithm of a string commitment scheme. For an instance* c*, a proof of set membership with respect to commitment scheme* C *and set* Φ *is a proof of knowledge for the following statement:*

$$PK\{(\sigma, \rho) : c \leftarrow \text{Com}(\sigma;\ \rho) \wedge \sigma \in \Phi\}$$

Remark: The proof system is defined with respect to *any* commitment scheme. Thus, in particular, if Com is perfectly hiding, then the language L_R consists of all commitments c (assuming that R is non-empty). Thus, for soundness, it is important that the protocol is a proof of knowledge. Furthermore, the statement being proven is the ability to open a commitment to an element contained in a public set. That is different from the claim that a commitment contains an element from a given public set, as the latter case gives no warranty that the prover knows the element in the commitment. Last but not least, it is important to note that in this chapter, proofs of set membership are, in fact, interactive arguments. As there is an absence of restriction on the commitment scheme used, a commitment scheme that is not perfectly binding (see Definition 2.32) will yield a proof system that is only computationally sound, in other words: an argument (see Section 2.3.2). The computationally bounded prover will know only one way of opening his commitment and cannot deduce other ways. As prior and present works refer to the problem as a “proof”, this term will be used here.

In order to solve the interactive set membership proof problem, we present three honest verifier zero-knowledge solutions. The restriction to honest verifiers can be explained on two grounds. Firstly and as explained in Section 2.4.1, there is a standard technique proposed by Cramer, Damgård, and MacKenzie in [CDM00] that can be used to transform an honest verifier zero-knowledge proof system into a general zero-knowledge one. Moreover, this technique is perfectly adapted to the special Σ protocols presented hereafter to solve the set membership proof problem. Secondly, the majority of other proof techniques are usually presented as honest verifier protocols. This is especially the case for range proofs. As one of the range proofs presented in Chapter 4 is based on the following set membership proofs, having the protocols in the honest verifier model allows for more accurate comparisons.

The first and main solution for set membership proof is based on the Boneh-Boyen signatures [BB04]. It requires that the *strong Diffie-Hellman assumption* holds. The second solution intends to show that other signature schemes can be used instead of the Boneh-Boyen signatures. We illustrate this with the Camenisch-Lysyanskaya signature scheme [CL02b]. In this second solution, the *strong RSA assumption* is needed instead of the strong Diffie-Hellman assumption. Note that the use of signature schemes for set membership proofs is a complete novelty introduced in [CCs08]. The third solution aims at replacing signature schemes in our set membership proof protocol with *cryptographic accumulators*. We use the cryptographic accumulator presented by Camenisch and Lysyanskaya [CL02a], which is also based on the *strong RSA assumption*.

Non-interactive set membership proof solutions can be obtained from their interactive versions by using standard techniques, such as for example the Fiat-Shamir heuristic [FS86]. These techniques often require an additional computational hardness assumption or an extended model in order to prove their security. For instance, the Fiat-Shamir heuristic requires the random oracle model, which assumes the existence of random oracles. Under this assumption, the Fiat-Shamir heuristic has been proven secure against chosen message attacks, by Pointcheval and Stern [PS96]. However, although this model enables the obtention of efficient protocols, it has been shown in [CGH98, CGH04] that some protocols in the random oracle model become insecure in the plain model. Moreover, Goldwasser and Tauman explained in [GK03] that some digital signatures obtained after a Fiat-Shamir transformation are simply insecure. A better alternative to the Fiat-Shamir heuristic would be the Lindell transform [Lin15] or the CPSV transform [CPSV16], where zero-knowledge is achieved in the standard model, and only the soundness requires the use of random oracles. In the case that random oracles need to be completely avoided, a solution based on the Groth-Sahai method [GS08, GS12a] could be used.

3.2 Prior and Related Work

The first concerns regarding membership proofs appeared when Ohta, Okamoto, and Koyama started to look into the problem of membership authentication in [OOK90]. In their paper, the focus of membership targets the authentication of a user to a privilege group, without revealing the identity of the user. To prove their membership status in a given group, all users of the group are provided with the same secret. This has been pointed out by Shu, Matsumoto, and Imai in [SMI91], where they provided a solution based on the discrete logarithm problem and on the difficulty of extracting modular roots. However their solution requires, in terms of the security parameter, a linear amount of exponentiations for both provers and verifiers, as well as a linear amount of transmitted elements. A similar protocol has been proposed later by Damgård and Jurik in [DJ01], where set membership proof is achieved by showing that a ciphertext encodes one valid plaintext from a given set of plaintexts. Their protocol is based on a generalization of the encryption scheme of Paillier [Pai99]. Hence the encryption of a plaintext $\sigma \in \mathbb{Z}_{n^s}$ is achieved with $C = E(\sigma, r) = g^{\sigma} r^{n^s} \pmod{n^{s+1}}$, where n is an RSA modulus, s is a natural number, g is a generator of an n^s order group $\mathbb{G}$, and r is a random element from a group $\mathbb{H}$ isomorphic to $\mathbb{Z}_n^*$. To show that C is the encryption of the plaintext $\sigma \in \Phi$, a prover performs a proof of knowledge that one of the elements $u_i = C \cdot g^{-i}$ is an $(n^s)^{\text{th}}$ power, where $i \in \Phi$. However, both prover and verifier need to perform $O(|\Phi|)$ exponentiations and the communication complexity is also $O(|\Phi|)$ group elements.

De Mare and Wright provided an alternative solution for set membership proofs in [dMW06], based on the hardness of the boolean 3-satisfiability (3SAT) problem. However, the elements composing the set are not public and the set size is restricted to, at most, up to a hundred elements.

Regarding subsequent results, Bayer and Groth provided, in [BG13, Bay13], a construction for set membership and set non-membership proofs without relying on either a trusted third party, or on signing the set elements. This however comes with a price in terms of both communication and computation. For a set Φ of size $|\Phi|$, their argument requires a communication complexity of $O(\log|\Phi|)$ group elements. As for the computational complexity, both provers and verifiers need to compute $O(\log|\Phi|)$ exponentiations. Regarding non-membership proofs, Blazy, Chevalier, and Vergnaud recently proposed, in [BCV15], a non-interactive argument based on the decisional Diffie-Hellman assumption.

Another subsequent result is provided by Canard et al. in [CCJT13], where they proposed a non-interactive set membership proof without any security proof, based on the Fiat-Shamir heuristic. Their scheme is based on a threshold variant of the ElGamal encryption [ElG84] and on a new variant of Boneh-Boyen signatures [BB04] that does not require pairing computations. Regarding the computational complexity of the prover, their set membership proof requires 15 exponentiations. Their communication complexity, despite being composed of a single message, is larger by a factor of 5. Moreover they require at least 2, and at most 3, verifiers. The computational complexity of verifiers is smallest when there are only 2 verifiers, although

it would still require 21 exponentiations. If the verifiers collude however, they will be able to decrypt the ciphertext containing a signature on the secret element of the prover, and hence to deduce his secret element from the uniqueness of the signature. To address these issues, Arfaoui et al. [ALT+15b, ALT+15a] worked on a similar version, also based on the random oracle model, without the use of encryption. Their protocol can be seen as a special case of the general signature based set membership proof mentioned in Section 5.1 of [CCs08], which is explained in Protocol 3.3. Their solution is to simply use the variant of Boneh-Boyen signatures presented in [CCJT13] in conjunction with Protocol 3.3. Compared to the Boneh-Boyen signature based set membership proof presented in Section 3.3, they achieve exactly the same communication complexity but gain in terms of prover and verifier computational complexity. Where the verifier is issuing the signatures, he needs to compute 4 additional exponentiations instead of 2 pairings, while the prover needs to perform 5 additional exponentiations instead of one pairing. Moreover, if the verifier is not the one issuing the signatures, the computational complexity gain for the prover remains the same. In this case, the verifier needs to do an additional exponentiation for one less pairing.

Benaloh and de Mare introduced, in [BdM93], the notion of *cryptographic accumulators*, based on the strong RSA assumption. Their use is to merge a set of elements into a single short accumulator, as well as producing a witness for each element proving that it has indeed been integrated into the accumulator. In [BdM93], Benaloh and de Mare proposed a membership testing where the secret choice is revealed. The exact same issue affects [BP97] by Barić and Pfitzmann. This issue was first solved by Camenisch and Lysyanskaya in [CL02a], with the introduction of *dynamic accumulators*, also under the strong RSA assumption. Dynamic accumulators not only allow the addition of elements into the accumulator, but also the deletion of elements. However, restrictions apply on the elements that can be accumulated. For instance, only prime numbers can be accumulated and the largest value has to be strictly smaller than the square of the smallest value.

Further work has been accomplished in order to improve accumulators ([San99, GTH02, TX03, Ngu05, AWSM07, LLX07, WWP07, PTT08, DT08, GTH09, Lip12b, FLZ14]) such as integrating composite numbers [TX03], providing non-membership protocols [LLX07, DT08, Lip12b, FLZ14], improving efficiency [GTH02, GTH09, PTT08], or using other computational hardness assumptions such as the q-strong Diffie-Hellman assumption [Ngu05]. However, all of these schemes require the secret element to be revealed in order to achieve the set membership proof.

Guo et al. proposed in [GMSV13] a related method called membership encryption. They hide the set description and attributes in a privacy preserving token $P(\mathbb{G})$. Their encryption method is performed on a public element x and the token $P(\mathbb{G})$. Decryption is then possible only if the user is holding the membership assertion $x \in \mathbb{G}$. Guo et al. claimed that their method could be used to achieve the set membership proof of a secret element in a public set, however their method would leak the value of the secret element.

Buhrman et al. in [BMRV00], Radhakrishnan et al. in [RSV02], Ostrovsky et al. in [ORS04], Kate et al. [KZG10] and Garg et al. in [GR15] focus on a different aspect of set membership proofs. In their papers, they consider different representations of sets in order to efficiently show the membership of a public element x into these sets. Hence they aim at answering questions of the form "is the public element x contained in the private set Ψ". Similarly, Micali, Kilian, and Rabin [MRK03] assess the problem in which a polynomial-time prover wants to commit to a finite secret set Ψ of strings so that, later on, he can, for any string x, reveal with a proof whether $x \in \Psi$ or $x \notin \Psi$ without leaking any knowledge beyond the membership assertions. In particular, the proofs do not reveal the elements nor the size of Ψ. Their solution is non-interactive and based on the computational hardness assumption of the *discrete logarithm problem*.

A particular aspect of set membership proofs appears when the set is a range of consecutive integer elements. This case is handled by range proofs rather than set membership proofs, as special techniques can be applied in order to increase efficiency. This matter will be explained in chapters 4 and 5.

Figure 3.1 provides some complexity comparisons between the protocols presented in this chapter and schemes from the literature with similar security goals. The asymptotical communication complexities are provided in terms of group elements. Assumptions based on the DLog problem require groups of size 256 bits. These groups will be denoted $\mathbb{G}_d$. Assumptions based on the factorization problem require groups of size 2048 bits. These groups will be denoted $\mathbb{G}_f$. The groups $\mathbb{G}_\mathrm{T}$ required for pairings are of size 3072 bits. Computational complexities are provided in terms of exponentiations (exp.) and pairings. Note that setup costs are here left aside, although they will be given later for the protocols presented in this thesis.

Schemes	*Communication*	*Computational*	
		Prover	*Verifier*
[SMI91]	$O(\|\Phi\|)$	$O(\|\Phi\|)$ exp.	$O(\|\Phi\|)$ exp.
[DJ01]	$O(\|\Phi\|)$	$O(\|\Phi\|)$ exp.	$O(\|\Phi\|)$ exp.
[BG13, Bay13]	$O(\log\|\Phi\|)$	$O(\log\|\Phi\|)$ exp.	$O(\log\|\Phi\|)$ exp.
[CCJT13]	$8\|\mathbb{G}_d\| + 9\|\mathbb{G}_f\|$	15 exp.	21 exp.
[ALT+15a]	$6\|\mathbb{G}_d\| + \|\mathbb{G}_\mathrm{T}\|$	8 exp.	7 exp.
Protocol 3.1 ([BB04] signature based)	$6\|\mathbb{G}_d\| + \|\mathbb{G}_\mathrm{T}\|$	3 exp., 1 pairing	3 exp., 2 pairings
Protocol 3.2 ([CL02b] signature based)	$19\|\mathbb{G}_d\| + 6\|\mathbb{G}_f\|$	14 exp.	13 exp.
Protocol 3.4 ([CL02a] accumulator based)	$19\|\mathbb{G}_f\|$	18 exp.	15 exp.

Figure 3.1 – Complexity comparisons for set Φ

3.3 Boneh-Boyen Signature Based Set Membership Proof

Here we present a set membership proof protocol that is inspired by the oblivious transfer protocol presented by Camenisch, Neven, and shelat [CNs07]. The basic idea is that the verifier first sends the prover a signature on every element in the set Φ. The prover therefore receives a signature on the particular element σ to which C is a commitment. The prover then "blinds" this received signature and performs a proof of knowledge that she possesses a signature on the committed element. Notice that the communication complexity of this proof depends on the cardinality of Φ, in particular because the first message of the verifier contains a signature on every element in Φ. The rest of the protocol, however, requires only a constant number of group elements to be sent. The novelty of this approach is that the first verifier message can be re-used in other proofs of membership; indeed, this property is used to achieve the results for range proofs in Chapter 4.

Computational assumptions. The protocol in this section requires Pedersen commitments, symmetric[2] bilinear groups, associated computational hardness assumptions, as well as the q-Strong Diffie Hellman assumption (q-SDH, see Section 2.2.2), with $|\Phi| = q$. Note that the q-SDH assumption implies the DLog assumption. Let PG be a symmetric bilinear pairing group generator that on input 1^κ outputs descriptions of multiplicative cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_T$ of prime order p where $\|p\| = k = (2\kappa + \log_2 q)$. Let $\mathbb{G}_1^* = \mathbb{G}_1 \setminus \{1\}$ and let $g \in \mathbb{G}_1^*$. The generated groups are such that there exists an admissible symmetric bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, meaning that

- for all $a, b \in \mathbb{Z}_p$ it holds that $e(g^a, g^b) = e(g, g)^{ab}$;
- $e(g, g) \neq 1$;
- and the bilinear map is efficiently computable.

Boneh-Boyen signatures. The set membership proof presented in this section relies on the elegant Boneh-Boyen short signature scheme [BB04] which is explained in Section 2.4.5 and briefly recalled here. The signer's secret key is $x \in_R \mathbb{Z}_p^*$ and the corresponding public key is $y = g^x$. The signature on a message m is $s = g^{1/(x+m)}$. Verification is done by checking that $e(s,\ y \cdot g^m) = e(g, g)$. Let us also recall here the following unforgeability property of the Boneh-Boyen short signature [BB04], paraphrased below:

Lemma 3.1 ([BB04](Lemma 1, Section 3.1))
Suppose the q-Strong Diffie Hellman assumption holds in $(\mathbb{G}_1, \mathbb{G}_1)$. Then the basic Boneh-Boyen signature scheme is q-secure against an existential forgery under a weak chosen message attack.

[2] Note that asymmetric bilinear groups could and should be used in practice as explained in Section 2.1.3. The use of symmetric bilinear groups here is solely to help readers understand the protocol.

Informally, this lemma states that under some specific assumptions, Boneh-Boyen signatures are unforgeable. This property will be needed to prove the soundness of the set membership proof described in this section. The reader is briefly reminded that the q-Strong Diffie Hellman assumption holds in $(\mathbb{G}_1, \mathbb{G}_1)$, if an adversary has a negligible advantage in outputting a pair $\left(m,\ g^{1/(x+m)}\right)$, when given as input $\left(g,\ g^x, g^{x^2},\ \cdots,\ g^{x^q}\right)$, where g is a generator of $\mathbb{G}_1$. For a more detailed and formal definition, see Section 2.2.2. A weak chosen message attack on a signature scheme consists of retrieving signatures on chosen messages, that were queried by the adversary before seeing the signature scheme public key. Furthermore, the Boneh-Boyen q-security property of a signature scheme informally consists of the unforgeability property when the attacker is allowed to query a signing oracle for strictly less than q messages of his choice. Further details regarding the weak chosen message attack and q-security can be found in Section 2.4.5. Last but not least, recall that Cheon provided a warning on the hardness of the q-Strong Diffie Hellman assumption in [Che06]. His results state that the computational complexity of recovering the secret element x is $O\left(\sqrt{p/q}\right) = O\left(2^{(k-\log q)/2}\right)$ group operations, where p is the k-bit prime order of $\mathbb{G}_1$. Hence, as explained in Section 2.2.2, particular attention needs to be given to the choice of p and the value of q, unless the value q is small enough compared to p. In practice, the binary length of p is often greater than 256 bits, for a value of q smaller than 15 bits. This leads to a computational complexity of an attack strictly higher than 2^{120} group operations. Hence in order to obtain a 128 bit security (meaning that the computational complexity is higher than 2^{128} group operations), the restriction on the security parameter k is $k \geqslant 256 + \log_2 q$.

Protocol explanation. The Boneh-Boyen signature based set membership proof is depicted in Protocol 3.1. The common input includes the following elements: a description of $\mathbb{G}_1$ and $\mathbb{G}_T$, as provided by the pairing group generator PG for the Boneh-Boyen signature; two generators g and h of $\mathbb{G}_1$ for the Pedersen commitment; the public set $\Phi \subset \mathbb{Z}_{|\mathbb{G}_1|}$; and a Pedersen commitment C to one element in Φ. As $|\mathbb{G}_1| = p$, this implies that $\Phi \subset \mathbb{Z}_p$. The prover input additionally contains elements σ and r such that $C = g^\sigma h^r$ and $\sigma \in \Phi$.

The first message exchanged consists of the Boneh-Boyen public key y of the verifier together with signatures A_i on every element contained in Φ. Here, $x \in \mathbb{Z}_p^*$ is the Boneh-Boyen secret key of the verifier. Note that instead of the verifier signing the elements in Φ, a trusted third party can be employed. Moreover, picking $x \in \mathbb{Z}_p^*$ should be done such that $-x$ is not present in Φ, as it is impossible to produce a signature on $-x$. Indeed, an honest verifier would be required to compute $A_{-x} = g^{\frac{1}{x-x}}$. As for the malicious verifier, he would need to provide a signature A_{-x} such that $e(g,g) = e(A_{-x},\ y \cdot g^x)$ for $y = g^{-x}$. In all cases, the correctness of the public key y and signatures A_i should be checked by the prover (and by the verifier if a trusted third party generated them). It is important to note here that regarding the security proof, this first message can be considered as a setup cost, and therefore the rest of the protocol will appear as a Σ-protocol.

The second message, which is sent by the prover to the verifier, is a blinding V_P on the

signature of σ, achieved by an exponentiation with a random secret v. At each step, every element is checked for correctness, such as verifying that an element is in the correct group. However many of these checks are only necessary when compiling from the honest verifier zero-knowledge model to the full zero-knowledge proofs. As previously mentioned, only the honest verifier case is presented here. The standard checks are provided for the sake of completeness.

Common Input: g, h, a commitment C, and a set Φ.
Prover Input: σ, r such that $C = g^{\sigma} h^{r}$ and $\sigma \in \Phi$.

$P \xleftarrow{y,\{A_i\}} V$
- Verifier picks $x \in_R \mathbb{Z}_p^*$ such that $-x \notin \Phi$ and sends $y \leftarrow g^{x}$ and $A_i \leftarrow g^{\frac{1}{x+i}}$, $\forall i \in \Phi$.
- Prover checks that $y \in \mathbb{G}_1^*$, $A_i \in \mathbb{G}_1^*$ and that $e(g,g) \stackrel{?}{=} e(A_i,\ y \cdot g^{i})$, $\forall i \in \Phi$.

$P \xrightarrow{V_P} V$
- Prover picks $v \in_R \mathbb{Z}_p^*$ and sends $V_P \leftarrow A_{\sigma}^{v}$.
- Verifier checks that $V_P \in \mathbb{G}_1^*$.

Prover and Verifier run $\mathrm{PK}\{(\sigma, r, v) : C = g^{\sigma} h^{r} \ \wedge \ V_P = g^{\frac{v}{x+\sigma}}\}$

$P \xrightarrow{a,D} V$
- Prover picks $s, t, m \in_R \mathbb{Z}_p^*$ and sends $a \leftarrow e\left(V_P^{-s} g^{t}, g\right)$ and $D \leftarrow g^{s} h^{m}$.
- Verifier checks that $a \in \mathbb{G}_{\mathrm{T}}$ and $D \in \mathbb{G}_1^*$.

$P \xleftarrow{c} V$
- Verifier sends a random challenge $c \in_R \mathbb{Z}_p^*$.
- Prover checks that $c \in \mathbb{Z}_p^*$.

$P \xrightarrow{z_{\sigma}, z_{v}, z_{r}} V$
- Prover sends $z_{\sigma} \leftarrow s - \sigma c$, $z_{v} \leftarrow t - vc$, and $z_r \leftarrow m - rc$.
- Verifier checks that $z_{\sigma}, z_{v}, z_r \in \mathbb{Z}_p^*$, that $D \stackrel{?}{=} C^{c} h^{z_r} g^{z_{\sigma}}$ and that $a \stackrel{?}{=} e\left(V_P, y\right)^{c} \cdot e\left(V_P^{-z_{\sigma}} g^{z_{v}}, g\right)$

Protocol 3.1 – Set membership proof protocol for set Φ, based on Boneh-Boyen signatures

Once the verifier has received the blinded signature, the prover and verifier engage in a proof of knowledge that the blinded signature corresponds to the secret σ contained in the initial commitment. The prover selects three random parameters s, t, and m in $\mathbb{Z}_p^*$ that will be used to blind his secret elements σ, v, and r respectively.

The initial message of the proof of knowledge is then a commitment D on the secret randomness s used by the prover, and a group element $a \in \mathbb{G}_{\mathrm{T}}$ which is a bilinear pairing on the blinded signature V_P using the same secret randomness s committed in D. At the end of the protocol, a will allow the verifier to check that V_P contains a valid blinded signature. Note, in

addition, that $D \in \mathbb{G}_1^*$. If $D = 1$ then the prover would be able to retrieve the discrete logarithm of h in base g by outputting $\log_g h = -s/m \pmod{p}$.

After this step, the verifier challenges the verifier with the challenge c. The prover replies to the verifier with the elements z_σ, z_v, and z_r. These will allow the verifier to complete the proof of knowledge by verifying that the following two equations hold:

$$D \stackrel{?}{=} C^c h^{z_r} g^{z_\sigma} \tag{3.1}$$

$$a \stackrel{?}{=} e\left(V_P, y\right)^c \cdot e\left(V_P^{-z_\sigma} g^{z_v}, g\right). \tag{3.2}$$

Recall again that $q = |\Phi|$ is supposedly a small number (bellow 15 bits length) compared to p (above 256 bits length). The attack from Cheon [Che06] on the q-Strong Diffie-Hellman assumption states that the computational complexity of recovering the secret key x is of $O\left(\sqrt{p/q}\right)$ group operations, instead of $O\left(\sqrt{p}\right)$ group operations. Hence the computational complexity reduction of $O\left(\sqrt{q}\right)$ is, in our case, polynomially bounded. Nevertheless, in order for the computational complexity of recovering the secret key x to be higher than 2^κ group operations (which is often called a κ-bit security), the security parameter k should be $k \geqslant 2\kappa + \log_2 |\Phi|$.

Theorem 3.2

If the $|\Phi|$-Strong Diffie-Hellman assumption associated with a pairing generator PG *holds, then Protocol 3.1 is a zero-knowledge argument of set membership for the set Φ.*

Proof

To show that Protocol 3.1 is a zero-knowledge argument of set membership, three security properties need to be satisfied: the *completeness* of the protocol, the *special soundness* property, and the *special honest verifier zero-knowledge* property.

The *completeness* of the protocol follows by inspection. In particular, the two last equalities hold as follows. Recall that $a = e\left(V_P^{-s} g^t, g\right)$. Hence,

$$\begin{aligned}
e\left(V_P, y\right)^c \cdot e\left(V_P^{-z_\sigma} g^{z_v}, g\right) &= e(V_P, y)^c \cdot e(V_P, g)^{-z_\sigma} \cdot e(g, g)^{z_v} \\
&= e(V_P, g^x)^c \cdot e(V_P, g)^{-s+\sigma c} \cdot e(g, g)^{t-vc} \\
&= e(V_P, g)^{xc} \cdot e(V_P, g)^{-s} \cdot e(V_P, g)^{\sigma c} \cdot e(g, g)^t \cdot e(g, g)^{-vc} \\
&= e(V_P, g)^{xc+\sigma c} \cdot e(g, g)^{-vc} \cdot e(V_P, g)^{-s} \cdot e(g, g)^t \\
&= e(g^{\frac{v}{x+\sigma}}, g)^{(x+\sigma)c} \cdot e(g, g)^{-vc} \cdot e(V_P, g)^{-s} \cdot e(g, g)^t \\
&= e(g^v, g)^c \cdot e(g, g)^{-vc} \cdot e(V_P, g)^{-s} \cdot e(g, g)^t \\
&= e(g, g)^{vc} \cdot e(g, g)^{-vc} \cdot e(V_P, g)^{-s} \cdot e(g, g)^t \\
&= e(V_P, g)^{-s} \cdot e(g, g)^t \quad = \quad e\left(V_P^{-s} g^t, g\right) \quad = \quad a.
\end{aligned}$$

Similarly, as $D = g^s h^m$, the following holds:

$$\begin{aligned}
C^c h^{z_r} g^{z_\sigma} &= (g^\sigma h^r)^c \cdot h^{m-rc} \cdot g^{s-\sigma c} &&= g^{\sigma c} h^{rc} \cdot h^{m-rc} \cdot g^{s-\sigma c} \\
&= g^{\sigma c+s-\sigma c} \cdot h^{rc+m-rc} &&= g^s h^m \\
&= D.
\end{aligned}$$

The *special soundness* follows from the extraction property of the proof of knowledge and the unforgeability of the Boneh-Boyen signature. This extraction property and how its extractor works will be explained first. Finally, a demonstration will be provided, explaining that if a malicious prover P^* is able to convince a verifier, then the extractor using this prover P^* can either be used to break the unforgeability property of the Boneh-Boyen signature (Lemma 3.1), or to create $\sigma \in \Phi$ and r such that $C = g^{\sigma} h^{r}$.

The extraction property of the proof of knowledge implies that for any prover P^* that convinces V with probability $\mathscr{S}$, there exists an extractor which interacts with P^* and outputs a witness (σ, r, v)within an expected number of steps bounded by $\frac{q(\kappa)}{\mathscr{S}-\mu(\kappa)}$, where μ is the knowledge error, q is a positive non zero polynomial, and κ is the security parameter. Moreover, following standard techniques ([BG92, Gol01]), the extractor obtains two related accepting transcripts tr and tr', for different challenges $c \neq c'$ but with the same initial elements $\{y, \{A_i\}, V_P, a, D\}$:

$$\begin{aligned} tr &= \{y, \{A_i\}, V_P, a, D, c, z_\sigma, z_v, z_r\}, \\ tr' &= \{y, \{A_i\}, V_P, a, D, c', z'_\sigma, z'_v, z'_r\}. \end{aligned}$$

Then, the witness can be obtained by computing:

$$\sigma = \frac{z_\sigma - z'_\sigma}{c' - c}; \quad r = \frac{z_r - z'_r}{c' - c}; \quad v = \frac{z_v - z'_v}{c' - c};$$

and its correctness can be confirmed with the following checks:

$$C \stackrel{?}{=} g^{\sigma} h^{r}; \quad V_P \stackrel{?}{=} A_{\sigma}^{v}.$$

The extractor succeeds since $(c' - c)$ is invertible in $\mathbb{Z}_p$.

If a malicious prover P^* is able to convince verifiers, then P^* can be (almost) directly used to mount a weak chosen-message attack against the Boneh-Boyen signature scheme.Indeed, the attacker will first learn all of the signatures of the elements in Φ. Then, as P^* has succeeded in convincing V, the extractor will output the witness (σ, r, v) by interacting with P^*, for $V_P = g^{\frac{v}{x+\sigma}}$ and $C = g^{\sigma} h^{r}$. Hence, if $v \neq 0$ (as shown below) then $V_P^{(1/v)}$ is a valid signature of σ. Due to the unforgeability property of the Boneh-Boyen signature scheme, the extractor outputs $\sigma \in \Phi$ and r such that $C = g^{\sigma} h^{r}$.

The following proof by contradiction, shows that v is a non zero element and hence invertible in $\mathbb{Z}_p^*$. Recall that a valid transcript necessarily satisfies the verification equation (3.2). Hence, the related transcripts tr and tr', used by the extractor, satisfy the equalities:

$$\begin{aligned} a &= e\left(V_P, y\right)^{c} \cdot e\left(V_P^{-z_\sigma} g^{z_v}, g\right), \text{ and} \\ a &= e\left(V_P, y\right)^{c'} \cdot e\left(V_P^{-z'_\sigma} g^{z'_v}, g\right). \end{aligned} \tag{3.3}$$

The proof by contradiction shows that if $c \neq c'$ and $v = 0$, then the extracted σ will fail to pass

the verification equation for the correctness of its signature:

$$e(g,g) \stackrel{?}{=} e(A_\sigma,\ y \cdot g^\sigma).$$

As the extractor is assumed to have obtained two related transcripts tr and tr', for different challenges $c \neq c'$, the following holds:

$$\begin{aligned}
v = 0 &\implies \frac{z_v - z'_v}{c' - c} = 0 \\
&\implies z_v = z'_v && (3.4)\\
&\implies e\left(V_P, y\right)^c \cdot e\left(V_P^{-z_\sigma} g^{z_v}, g\right) = e\left(V_P, y\right)^{c'} \cdot\ e\left(V_P^{-z'_\sigma} g^{z_v}, g\right) && (3.5)\\
&\implies e(V_P, g^x)^c \cdot\ e(V_P, g)^{-z_\sigma} = e(V_P, g^x)^{c'} \cdot\ e(V_P, g)^{-z'_\sigma} && (3.6)\\
&\implies e(V_P, g)^{xc - z_\sigma} = e(V_P, g)^{xc' - z'_\sigma} \\
&\implies xc - z_\sigma = xc' - z'_\sigma \pmod p && (3.7)\\
&\implies x(c - c') = z_\sigma - z'_\sigma \pmod p && (3.8)\\
&\implies \sigma = \frac{z_\sigma - z'_\sigma}{c' - c} = -x \pmod p && (3.9)
\end{aligned}$$

Due to equation (3.4), z'_v is replaced by z_v in equation (3.3) to obtain equation (3.5). As $e(g,g) \neq 0$ by definition ($\mathbb{G}_\mathrm{T}$ is a multiplicative group), equation (3.5) can be divided by $e(g,g)^{z_v}$ to obtain equation (3.6). Since $V_P \neq 1$ we have $e(V_P, g) \neq 1$, which implies equation (3.7). Furthermore, as $c \neq c'$, equation (3.8) can be divided by $(c' - c)$. As a result, equation (3.9) will fail the check $e(g,g) \stackrel{?}{=} e(A_\sigma,\ y \cdot g^\sigma)$, for any signature $A_\sigma \in \mathbb{G}_1^*$.

1. Sim retrieves $y, \{A_i\}$ from V^* (or from a trusted third party).
2. Sim chooses $\sigma \in_R \Phi$, $v \in_R \mathbb{Z}_p^*$ and computes $V_P \leftarrow A_\sigma^v$.
3. Sim runs the simulator of PK$\{(\sigma, r, v) : C = g^\sigma h^r \ \wedge\ V_P = g^{\frac{v}{x+\sigma}}\}$.
   (a) On challenge $c \in \mathbb{Z}_p^*$, Sim chooses $z_\sigma, z_v, z_r \in_R \mathbb{Z}_p^*$.
   (b) Finally, Sim computes $a \leftarrow e(V_P, g)^{-z_\sigma - \sigma c} e(g,g)^{z_v + vc}$ and $D \leftarrow C^c h^{z_r} g^{z_\sigma}$.
4. Sim returns the transcript $\{y, \{A_i\}, V_P, a, D, c, z_\sigma, z_v, z_r\}$.

Figure 3.2 – Simulator for the set membership proof protocol

Finally, to prove *special honest verifier zero-knowledge*, we construct a simulator Sim for any verifier V^*, as depicted in Figure 3.2. The goal of the simulator Sim is to simulate all possible interactions with any honest prover P. Sim will first follow the initialization and the blinding instructions honestly, using a random $\sigma \in_R \Phi$ and a random $v \in_R \mathbb{Z}_p^*$ to compute V_P. Then Sim runs the simulator of the Σ-protocol PK$\left\{(\sigma, r, v) : C = g^\sigma h^r \ \wedge\ V_P = g^{\frac{v}{x+\sigma}}\right\}$. Hence, on the challenge $c \in \mathbb{Z}_p^*$, the simulator first picks z_σ, z_v, z_r in $\mathbb{Z}_p^*$ randomly, and then computes a, D

as follows:

$$a = e(V_P, g)^{-z_\sigma - \sigma c} e(g, g)^{z_\upsilon + \upsilon c},$$
$$D = C^c h^{z_r} g^{z_\sigma}.$$

The output of the simulator is a transcript $\{y, \{A_i\}, V_P, a, D, c, z_\sigma, z_\upsilon, z_r\}$, which has an identical probability distribution to a normal transcript between regular provers and verifiers. We can easily see that y, $\{A_i\}$ and c are identical in both transcripts, as they are provided by the verifier. V_P has the same probability distribution as it is computed with a valid $\sigma \in \Phi$ and a random $\upsilon \in_R \mathbb{Z}_p^*$. As s, t, m are randomly picked in $\mathbb{Z}_p^*$, they impose the same randomness towards $z_\sigma, z_\upsilon, z_r$. Hence $z_\sigma, z_\upsilon, z_r$ also have the same probability distribution. It is straightforward to see that D has the same probability distribution as it is computed from the same elements with the same distributions. Last but not least, as $s = z_\sigma + \sigma c$ and $t = z_\upsilon + \upsilon c$, a has the same probability distribution for the same reasons as for D. Since $\mathbb{G}_1$ is a prime-order group, then the blinding is perfect in the first two steps of the simulator; thus the zero-knowledge property follows from the zero-knowledge property of the Σ-protocol in the third step. ■

Communication and Computational Complexity. As the first message of Protocol 3.1 can be regarded as a setup procedure, it will not be included in the complexity analysis. Nevertheless, its cost is mentioned here for comparison purposes. The first message consists of $|\Phi|$ signatures and the public key y, which sum up to $|\Phi| + 1$ group elements for the communication, $|\Phi| + 1$ exponentiations for the verifier (or the trusted third party), and $|\Phi| + 1$ bilinear pairings for the prover in the non-honest verifier model.

Overall, the communication complexity of Protocol 3.1 consists of 2 group elements in $\mathbb{G}_1$, 1 group element in $\mathbb{G}_T$ and 4 elements in $\mathbb{Z}_p^*$. Regarding computational complexity, the honest verifier setting is assumed. Hence, the prover computational cost is dominated by 3 exponentiations and 1 pairing. The verifier computational cost is dominated by 3 exponentiations and 2 pairings.

3.4 Alternative Signature Based Set Membership Proof

The set membership proof protocol presented in Section 3.3 makes the verifier produce signatures on the set elements, send them to the prover, and then requires the prover to show that he knows a signature (from the verifier) and the element he holds. In other words, this last step requires the prover to be able to prove the knowledge of a signature on a value that he has committed to, using the Pedersen commitment scheme. Concretely, the weak signature scheme by Boneh and Boyen is employed. The following is a discussion on alternative signature schemes which allow the whole protocol to be based on different assumptions. Apart for the weak Boneh-Boyen signature scheme, there are other signature schemes that could be employed.

In terms of assumptions, one notable alternative would be the one by Camenisch and Lysyanskaya [CL02b] that is based on the strong RSA assumption. It is not hard to adapt the protocol given in Section 3.3 to that signature scheme, in particular as Camenisch and Lysyanskaya gave protocols to prove knowledge of a committed value in their paper [CL02b]. It should further be mentioned that Pointcheval and Sanders recently developed an improvement of [CL02b] in [PS15], where signatures consists of two group elements instead of three, with the help of specific computational assumptions (notably the LRSW assumption [LRSW99]). However, the solution presented in Section 3.3 remains the most efficient one. The alternatives discussed in this section are of similar efficiency.

Computational assumptions. The protocol in this section requires the notion of quadratic residues modulo n, special RSA modulus, the strong RSA assumption, and the Fujisaki-Okamoto commitment scheme ([FO98, DF02], see Section 2.4.2). An RSA modulus $n = pq$ is called special if p and q are both safe primes. Hence $n = (2p'+1)(2q'+1)$, where p' and q' are both Sophie-Germain primes. The strong RSA assumption is explained in Section 2.2.2. As a brief reminder, this assumption states that for a random $a \in \mathbb{Z}_n^*$, it is hard to compute $e \geqslant 3$ and the e^{th} root of a, where n is an RSA modulus. Recall in addition that for the commitment, instead of performing computations in a group of prime order as for a Pedersen commitment, the Fujisaki-Okamoto commitment uses the set QR_n of quadratic residues modulo a special RSA modulus n.

Camenisch-Lysyanskaya signatures. The set membership proof in this section relies on the Camenisch-Lysyanskaya signature scheme in [CL02b] which is explained hereafter. The signer's secret key is a safe prime $\tilde{p}$ such that $\tilde{n} = \tilde{p}\tilde{q}$ is a special RSA modulus of binary length $2k$, where k is the general security parameter. The corresponding public key is $(\tilde{n}, a, b, c)$, where $a, b, c \in_R QR_{\tilde{n}} \setminus \{1\}$[3]. The signature on a message m of binary length ℓ_m, is a tuple (s, e, v) such that $v^e \equiv a^m b^s c \pmod{\tilde{n}}$, where e is a random prime number of binary length $\ell_e \geqslant \ell_m + 2$, s is a random number of binary length $\ell_s > 2k + \ell_m$, and where v is obtained with $v = (a^m b^s c)^{e^{-1}} \pmod{\tilde{n}}$. To verify a signature (s, e, v) on a message m against a public key $(\tilde{n}, a, b, c)$, a verifier runs the predicate $\mathsf{Verify}_{(\tilde{n},a,b,c)}(m, s, e, v)$ which consists of checking that $v^e \equiv a^m b^s c \mod \tilde{n}$. Camenisch and Lysyanskaya suggested the following values for the security parameters: $k = 512$, $\ell_m = 160$, $\ell_e = 162$, $\ell_s = (2k + \ell_m + 160) = 1344$. Let us also recall the security property of the Camenisch-Lysyanskaya signature scheme [CL02b], paraphrased below:

Theorem 3.3 ([CL02b](Theorem 1))

The Camenisch-Lysyanskaya signature scheme is secure under the strong RSA assumption. More precisely, if a forger breaks the signature scheme in time $p(k)$ with probability $\epsilon(k)$, then the strong RSA assumption can be broken in time $O(p(k))$ with probability $\Omega(\epsilon(k)/p(k))$.

[3] Note that the requirement of removing the element 1 from $QR_{\tilde{n}}$ is missing in the original version [CL02b].

Protocol explanation. The Camenisch-Lysyanskaya signature based set membership proof is depicted in Protocol 3.2. The common input includes the following elements: the public parameters of a Fujisaki-Okamoto commitment (a safe RSA modulus $n = pq$ and two generators $g, h \in QR_n$); the public set Φ of elements with binary length smaller than ℓ_m; a Fujisaki-Okamoto commitment C to one element in Φ and security parameters ℓ_e, ℓ_s. The security parameters ℓ_e, ℓ_s are used to define the length of the signature elements e_i, s_i respectively. The prover input additionally contains an element $\sigma \in \Phi$ and $r \in \mathbb{Z}^*_{|\mathbb{G}|}$ such that $C = g^\sigma h^r$.

Common Input: g, h, a commitment C, a set Φ,
and security parameters ℓ_m, ℓ_s, ℓ_e.

Prover Input: σ, r such that $C = g^\sigma h^r$ and $\sigma \in \Phi$.

$P \xleftarrow{(\tilde{n},a,b,c),\ \{(s_i,e_i,v_i)\}} V$

- Verifier picks two safe primes $\tilde{p}, \tilde{q}$, sets $\tilde{n} \leftarrow \tilde{p}\tilde{q}$,
  picks $a, b, c \in_R QR_{\tilde{n}} \setminus \{1\}$,
  picks random numbers s_i of length ℓ_s, $\forall i \in \Phi$,
  picks random primes $e_i \in_R \left(2^{\ell_e - 1}, 2^{\ell_e}\right)$, $\forall i \in \Phi$,
  computes $v_i \leftarrow \left(a^i b^{s_i} c\right)^{1/e_i}$, $\forall i \in \Phi$,
  sends $(\tilde{n}, a, b, c)$ and (s_i, e_i, v_i), $\forall i \in \Phi$.
- Prover checks for every $i \in \Phi$ that $e_i \in_R \left(2^{\ell_e - 1}, 2^{\ell_e}\right)$ and that $v_i^{e_i} \equiv a^i b^{s_i} c$.

Prover and Verifier run

$$\mathrm{PK}\{(\sigma, r, s_\sigma, e_\sigma, v_\sigma) : C = g^\sigma h^r \ \wedge\ \mathsf{Verify}_{(\tilde{n},a,b,c)}(\sigma, s_\sigma, e_\sigma, v_\sigma) = 1\}$$

Protocol 3.2 – Set membership proof protocol for set Φ,
based on Camenisch-Lysyanskaya signatures

The first message exchanged consists of the Camenisch-Lysyanskaya signature public key $(\tilde{n}, a, b, c)$ of the verifier, together with a signature (s_i, e_i, v_i) on every element of $i \in \Phi$. The secret key of the verifier is the safe prime $\tilde{p}$ which allows him to factorize $\tilde{n} = \tilde{p}\tilde{q}$ into two safe primes. Obviously, $(\tilde{p}, n)$ should be coprime as well as $(\tilde{q}, n)$, otherwise the owner of the factorization $n = pq$ would be allowed to produce signatures on behalf of the verifier, and hence to break the unforgeability property of the Camenisch-Lysyanskaya signature scheme. As in the case of Section 3.3 with the Boneh-Boyen signature scheme, standard checks should be performed such as verifying the correctness of signatures (s_i, e_i, v_i) on every element $i \in \Phi$.

Once the signatures are delivered to the prover and their correctness has been verified, the prover and the verifier engage in an honest verifier zero-knowledge proof of knowledge of a signature, such that the signature corresponds to the element hidden in the commitment

C of the prover. Camenisch and Lysyanskaya provided two protocols achieving this proof of knowledge of a signature in [CL02b], both based on the strong RSA assumption. Their first protocol is intended to be comprehensive rather than optimized, while the second one is focused on optimization. For the sake of completeness, the optimized protocol is detailed in Appendix A. Let us recall their security property, which is paraphrased from [CL02b]:

Lemma 3.4 ([CL02b] (Lemma 8 and lemma 15))
Assume that $C = g^{\sigma} h^{r}$ is a Fujisaki-Okamoto commitment on the element $\sigma \in \Phi$ with randomness $r \in \mathbb{Z}^{*}_{|\mathbb{G}|}$, as defined in protocol 3.2. Let (s, e, v) be a Camenisch-Lysyanskaya signature on the element σ as defined in [CL02b]. Let Verify_{PK} be the verification algorithm of the Camenisch-Lysyanskaya signature scheme, where PK is the public key. The two Camenisch-Lysyanskaya protocols in [CL02b] for proof of knowledge of a signature are zero-knowledge proofs of knowledge of the values (σ, r, s, e, v) such that $\mathsf{Verify}_{PK}(\sigma, s, e, v) = 1$.

Theorem 3.5
If the strong RSA assumption holds, then Protocol 3.2 is a zero-knowledge argument of set membership for the set Φ.

Proof
Recall that the underlying proof of knowledge

$$\mathrm{PK}\{(\sigma, r, s_{\sigma}, e_{\sigma}, v_{\sigma}) : C = g^{\sigma} h^{r} \ \wedge \ \mathsf{Verify}_{(\tilde{n},a,b,c)}(\sigma, s_{\sigma}, e_{\sigma}, v_{\sigma}) = 1\}$$

is detailed in Appendix A.

The completeness of the protocol follows from the completeness of the underlying proof of knowledge.

The special soundness property follows from the unforgeability of the Camenisch-Lysyanskaya signature scheme (Theorem 3.3) and from the extraction property of the Camenisch-Lysyanskaya proof of knowledge of a signature (Lemma 3.4). The goal of the extractor is to produce a valid signature on an opening σ of C, by invoking the underlying extractor of the proof of knowledge of a signature. Moreover, the extractor is actually identical to the one of the underlying proof of knowledge. Indeed, if the extractor of the proof of knowledge of a signature succeeds and outputs a witness $(\sigma, r, s_{\sigma}, e_{\sigma}, v_{\sigma})$, either it can be directly used to break the unforgeability of the Camenisch-Lysyanskaya signature scheme as $(s_{\sigma}, e_{\sigma}, v_{\sigma})$ would be a valid signature of σ, or the witness contains $\sigma \in \Phi$ and an opening (σ, r) of C. However, if this extractor fails, then a reduction can be made to break the strong RSA assumption.

In order to prove special honest verifier zero-knowledge, the simulator follows the initialization steps honestly, then invokes the simulator of the underlying honest verifier zero-knowledge proof of knowledge of a signature. ■

Communication and Computational Complexity. As in the case of Protocol 3.1, the first message of Protocol 3.2 can be regarded as a setup procedure, and thus will not be included

in the complexity analysis. Nevertheless, we mention its cost for comparison purposes. The first message consists of $|\Phi|$ signatures and the public key $(\tilde{n}, a, b, c)$. For the communication, this amounts to $4+|\Phi|$ elements in $\mathbb{Z}_{\tilde{n}}$, $|\Phi|$ elements of length ℓ_e and $|\Phi|$ elements of length ℓ_s. The verifier (or the trusted third party) will be required to perform $3|\Phi|$ exponentiations. The prover in the non-honest verifier model, will also be required to perform the same amount of exponentiations.

Overall, the complexity of Protocol 3.2 is identical to the Camenisch-Lysyanskaya proof of knowledge of a signature. The communication complexity sums up to 6 group elements and 9 elements in the size of the group order. Regarding computational complexity for the honest verifier setting, the prover computational cost is dominated by 14 exponentiations. The verifier computational cost is dominated by 13 exponentiations.

Common Input: a commitment C, a set Φ,
the commitment scheme parameters $\mathsf{Param}_{\mathsf{com}}$,
and the signature scheme parameters $\mathsf{Param}_{\mathsf{sign}}$.

Prover Input: σ, r such that $C = \mathsf{Commit}(\sigma, r)$ and $\sigma \in \Phi$.

$P \xleftarrow{Pub,\ \{A_i\}} V$

- Verifier picks his secret key Sk,
  generates the corresponding public key Pub,
  computes the signature $A_i = \mathsf{Sign}_{Sk}(i)$, $\forall i \in \Phi$,
  sends Pub and $\{A_i\}$, $\forall i \in \Phi$.
- Prover checks for every $i \in \Phi$ that
  $\mathsf{Verify}_{Pub}(i,\ A_i) = 1$.

Prover and Verifier run

$$\mathrm{PK}\{(\sigma, r, A) : C = \mathsf{Commit}(\sigma, r) \ \wedge \ \mathsf{Verify}_{Pub}(\sigma, A) = 1\}$$

Protocol 3.3 – Set membership proof protocol for set Φ,
based on a general signature scheme

Using Alternative Signature Schemes. The general idea of using a set membership proof based on a signature scheme consists of two steps. At first, the prover is given signatures on every element of the public set Φ. This allows the prover to select the signature corresponding to his secret element $\sigma \in \Phi$. The signatures can be provided by either the verifier or a trusted third party. In the second step, the prover runs a zero-knowledge proof of knowledge of a signature with the verifier, to ensure that the prover knows a valid signature on his secret element that he has previously committed to. The assumptions needed are inherited from the ones used in the commitment scheme, in the signature scheme, and in the proof of knowledge

needed. A general description is provided in Protocol 3.3, where $\mathsf{Param}_{\mathsf{sign}}$, Sign, and Verify refer respectively to the parameters, the signature algorithm, and the verification algorithm of the signature scheme. Pub and Sk are, respectively, the public key and the secret key of the signer. As for the commitment scheme used, $\mathsf{Param}_{\mathsf{com}}$ refers to the public parameters of the commitment, and the commit algorithm $\mathsf{Commit}(m, r)$ returns a commitment to the message m under randomness r. Note that depending on the commitment and signature scheme used, the proof of knowledge will force some restrictions on the message space provided by the public set Φ.

3.5 Accumulator Based Set Membership Proof

The reasons that signature schemes were employed in the previous two sections, is that the prover needed to show that he committed to a value for which he knows an authenticator without revealing that value or the authenticator. Now it turns out that exactly the same goal can be achieved with cryptographic accumulators with similar complexities.

Cryptographic accumulators are briefly recalled here, as they are explained in more detail in Section 2.4.6. A cryptographic accumulator is an algorithm that allows a user to compress a list of elements into a single accumulator value. For each element, there exists a witness attesting to the fact that the element is indeed contained in the accumulator value. For some cryptographic accumulators, there exist efficient proof systems that allow a prover holding an accumulated element and its corresponding witness to prove to a verifier in zero knowledge that he is privy to an element that is contained in the accumulator. Camenisch and Lysyanskaya have given details of such an accumulator in [CL02a], with the introduction of dynamic accumulators (see Section 2.4.6). They also provided a protocol that a committed value is contained in the accumulator based on the strong RSA assumption. However, their proof of knowledge needs to be modified in order to be used in a set membership proof protocol, as only primes in a restrictive integer range are allowed to be accumulated.

The idea of building an efficient set membership proof with dynamic accumulator is very similar to the signature based one. The verifier adds each element of the set into the accumulator and sends the accumulator value to the prover together with the corresponding witness for each element. The prover then proves to the verifier that the value he has committed to is contained in the accumulator produced by the verifier, by using the appropriate witness.

Computational assumptions. As in Section 3.4, the accumulator based set membership proof requires Pedersen commitments, the notions of quadratic residues, and special RSA modulus, as well as the accumulator associated computational hardness assumptions. In this case, the dynamic accumulators of Camenisch and Lysyanskaya are based on the strong RSA assumption.

Camenisch-Lysyanskaya accumulators. The dynamic accumulators of Camenisch and Lysyanskaya [CL02a] are briefly recalled here, as they are explained in more detail in Section 2.4.6. Assuming a special RSA modulus $\tilde{n} = (2\tilde{p}+1)(2\tilde{q}+1)$ of length k, the message space of the elements to be accumulate is the set of prime numbers e, such that $e \notin \{\tilde{p}, \tilde{q}\}$ and $2 < A \leqslant e < A^2$. These requirements come from the structure of the accumulator and the way in which elements are accumulated. Indeed, to add an element e into the accumulator $v \in QR_{\tilde{n}}^*$, the user computes $v' = v^e \pmod{\tilde{n}}$. The witness of an element e contained in an accumulator $v \in QR_{\tilde{n}}^*$ is the element $w_e = v^{1/e} \pmod{\tilde{n}}$. To check that an element e is indeed contained in the accumulator v, the following verification is performed: $v \stackrel{?}{=} (w_e)^e \pmod{\tilde{n}}$. Last but not least, we recall the security theorem of the Camenisch-Lysyanskaya dynamic accumulators [CL02a], paraphrased below:

Theorem 3.6 ([CL02a] (Theorem 2))
Under the strong RSA assumption, the Camenisch-Lysyanskaya accumulator scheme is a secure dynamic accumulator.

Common Input: g, h, a commitment C, a set Φ.
Prover Input: σ, r such that $C = g^\sigma h^r$ and $\sigma \in \Phi$.

$P \xleftarrow{\tilde{n},\, v,\, \tilde{g},\, \tilde{h},\, \Theta} V$

- Verifier picks a safe prime product $\tilde{n} = (2\tilde{p}+1)(2\tilde{q}+1)$,
  picks $u, \tilde{g}, \tilde{h} \in_R QR_{\tilde{n}}$,
  picks $a_i \in \{0,1\}^{\tilde{k}}$ such that $e_i = i \cdot 2^k + a_i$ are prime, $\forall i \in \Phi$,
  computes $v \leftarrow u^{2\prod e_i} \pmod{\tilde{n}}$; $w_i \leftarrow v^{1/e_i} \pmod{\tilde{n}}$, $\forall i \in \Phi$,
  sends $\tilde{n}$, v, $\tilde{g}$, $\tilde{h}$, and $\Theta \leftarrow \{(e_i, w_i) : i \in \Phi\}$.
- Prover checks the correctness of $\tilde{n}$, v, $\tilde{g}$, $\tilde{h}$, Θ.
- Prover and Verifier run PK$\{(\alpha) : \tilde{g} = \tilde{h}^\alpha \pmod{\tilde{n}}\}$.

$P \xrightarrow{W,\, R,\, C_e} V$

- Prover picks $r_1, r_2, r_e \in \mathbb{Z}_{\lfloor \tilde{n}/4 \rfloor}$,
  sends $W \leftarrow w_\sigma \tilde{h}^{r_1} \pmod{\tilde{n}}$, $R \leftarrow \tilde{g}^{r_1} \tilde{h}^{r_2} \pmod{\tilde{n}}$,
  and $C_e \leftarrow \tilde{g}^{e_\sigma} \tilde{h}^{r_e} \pmod{\tilde{n}}$.
- Verifier checks that W, R, $C_e \in QR_{\tilde{n}}$.

Prover and Verifier run

$$\begin{aligned} \text{PK}\{(\sigma, r, e_\sigma, a_\sigma, r_e, r_1, r_2) :\ & C = g^\sigma h^r \ \wedge\ C_e = \tilde{g}^{e_\sigma} \tilde{h}^{r_e} \pmod{\tilde{n}} \ \wedge \\ & C_e = (\tilde{g}^{2^k})^\sigma \tilde{g}^{a_\sigma} \tilde{h}^{r_e} \pmod{\tilde{n}} \ \wedge\ R^{e_\sigma} = \tilde{g}^{r_1 e_\sigma} \tilde{h}^{r_2 e_\sigma} \pmod{\tilde{n}} \ \wedge \\ & v = W^{e_\sigma} \tilde{h}^{-r_1 e_\sigma} \pmod{\tilde{n}} \ \wedge\ a_\sigma \in [-2^{k-1}, 2^{k-1}]\} \end{aligned}$$

Protocol 3.4 – Set membership proof protocol for set Φ, based on Camenisch-Lysyanskaya accumulators

Protocol explanation. One complication that needs to be dealt with here, is that the accumulator only allows the accumulation of prime numbers, whereas the set Φ is composed of arbitrary bits strings. A mapping thus need to be encoded. This can be done as follows: let Φ be our set, where the elements $i \in \Phi$ are assumed to be integers; let $e_i = i2^k + a_i$, where $a_i < 2^{\bar{k}} < 2^k$ is selected so that e_i is prime. The security parameter $\bar{k}$ defines the length of elements a_i. Moreover, it is required that $A^2 - 1 < q/2$, where q is the order of the Pedersen commitment group. This requirement is inherited from the requirements of the proof of knowledge that e_i is accumulated in v (see Appendix B and [CL02a]). With this encoding, the verifier can produce a proof of knowledge that e_σ corresponds to his committed element σ, and that e_σ is accumulated in v. Hence the Camenisch-Lysyanskaya accumulator based set membership proof depicted in Protocol 3.4 is obtained.

The common input includes the following elements: the description $\mathbb{G}$ of a Pedersen commitment group; two generators g and h of $\mathbb{G}$ for the Pedersen commitment; the public set Φ ; and a Pedersen commitment C to one element in Φ. The prover input additionally contains the elements $\sigma \in \Phi$ and $r \in \mathbb{Z}^*_{|\mathbb{G}|}$ such that $C = g^\sigma h^r$.

The first step consists of a message sent by the verifier to the prover, followed by a small proof of knowledge. This first message consists of the accumulator v, public parameters $(\widetilde{n}, \widetilde{g}, \widetilde{h})$ for the accumulator, and a set Θ regrouping the witnesses w_i for the accumulated element e_i corresponding to the elements $i \in \Phi$. Furthermore, the verifier possesses the primes $\widetilde{p}$ and $\widetilde{q}$ that decompose $\widetilde{n} = (2\widetilde{p}+1)(2\widetilde{q}+1)$, as well as the element u used to generate the accumulator v. As was the case for the non-honest verifier model, standard checks should be performed such as verifying the correctness of the elements $\widetilde{n}$, v, $\widetilde{g}$, $\widetilde{h}$, Θ. The proof of knowledge that follows aims at convincing the prover that $\widetilde{g} \in \langle \widetilde{h} \rangle$. The straightforward way to achieve this, is that the prover runs the proof of knowledge $\mathrm{PK}\{(\alpha) : \widetilde{g} = \widetilde{h}^\alpha \pmod{\widetilde{n}}\}$ with the verifier using binary challenges. Another, more efficient, way is described by Bangerter et al. [BCM05], based on the work of Cramer [Cra97].

The second step, is the core of the set membership proof. A reply is provided from the prover to the verifier, that consists of a blinding W on the witness w_σ, and additionally of two commitments R, C_e. The commitment R is a commitment on the randomness used for the blinding W. The commitment C_e is a commitment on the prime e_σ corresponding to the committed element σ. In both this step and the previous one, every element is checked for correctness, such as verifying that an element is in the correct group. However, these checks are only necessary when compiling from the honest verifier zero-knowledge model to the full zero-knowledge proofs. Here again, only the honest verifier case is of concern. The standard checks are provided for the sake of completeness.

The last stage of this second step, is a proof of knowledge that e_σ corresponds to the σ in the initial commitment of the prover, and that e_σ is also contained in the accumulator. The resulting proof of knowledge is given in Appendix B, where the accumulator proof given by Camenisch and Lysyanskaya [CL02a] is adapted to this setting. This adaptation mainly

concerns the fact that the correspondence between e_σ and the committed σ needs to be additionally proven to hold. For this to work, the prover needs to show that $e_\sigma = \sigma 2^k + a_\sigma$ holds, for some a_σ known to the prover. Here it is, of course, important that this a_σ be at most of length $\bar{k} < k$ bits. This can be enforced efficiently, provided that $\bar{k}$ is a couple of bits smaller than k, where in practice the difference should be about 300 bits for this to work. More precise accuracy could be achieved with the range proofs presented in Chapter 4 and in Chapter 5. However, for this specific purpose, they would be less efficient.

Theorem 3.7

If the strong RSA assumption holds, then Protocol 3.4 is a zero-knowledge argument of set membership for the set Φ.

Proof

The completeness of the protocol follows from the completeness of the underlying proofs of knowledge.

The special soundness property follows from the security property of the Camenisch-Lysyanskaya accumulator scheme (Theorem 3.6) and from the extraction property of the proof of knowledge of a committed accumulated element described in Appendix B. This extraction property is directly derived from the extraction property of the Camenisch-Lysyanskaya proof of knowledge that a committed value is accumulated ([CL02a], Theorem 3). The extractor goal is to produce a valid pair (e_σ, w_σ) on σ, such that w_σ is a witness that e_σ has been accumulated in v, and $e_\sigma = \sigma 2^k + a_\sigma$, where $a_\sigma \in [-2^{k-1}, 2^{k-1}]$. This is achieved by invoking the underlying extractor for the proof of knowledge of a committed accumulated element, as this extractor directly provides the necessary elements. If $\sigma \notin \Phi$, this can be directly used to break the security property of the Camenisch-Lysyanskaya accumulator scheme as w_σ would be a valid witness for a prime e_σ that has not been accumulated. However, if the extractor fails, then a reduction can be made to break the strong RSA assumption.

To prove special honest verifier zero-knowledge, the simulator follows the first step honestly. It then randomly selects $\sigma \in_R \Phi$, honestly computes and sends W, R, C_e to the verifier V^*, and invokes the simulator of the underlying honest verifier zero-knowledge proof of knowledge of a committed accumulated element. ■

Communication and Computational Complexity. As the first message of Protocol 3.4 can be regarded as a setup procedure, it will not be included in the complexity analysis. Nevertheless, its cost is mentioned for the purpose of comparison. The first message consists of the set Θ, the accumulator v and public parameters $(\tilde{n}, \tilde{g}, \tilde{h})$. For the communication, these elements can be approximated with an upper bound of $4 + 2|\Phi|$ elements in $\mathbb{Z}_{\tilde{n}}$. The verifier will be required to perform $1 + |\Phi|$ exponentiations for the witnesses and the accumulator computation. The prover in the non-honest verifier model will be required to perform $|\Phi|$ exponentiations for checking the correctness of the witnesses. Note also, that for many applications, the parameters $\tilde{n}$, v, $\tilde{g}$, $\tilde{h}$, and Θ only needs to be computed and published once (possibly by a trusted

third party). In this case the communication and computational complexity of Protocol 3.4 becomes independent of the number of elements in the set Φ.

With regard to the communication complexity, the exchanged elements have, at most, a length of k bits, where k is the security parameter defining the length of $\widetilde{n}$. Thus, instead of mentioning the precise size of each element, it will simply be implied that group elements have at most up to k bits of length. The communication complexity therefore includes 3 group elements for the proof of knowledge PK$\{(\alpha) : \widetilde{g} = \widetilde{h}^{\alpha} \pmod{\widetilde{n}}\}$, the three commitments W, R, C_e, and lastly, 13 group elements for the remaining proof of knowledge of a committed accumulated element. Overall, the communication complexity sums up to 19 group elements. Regarding computational complexity, the honest verifier setting is assumed. The prover computational cost is dominated by 18 exponentiations. The verifier computational cost is dominated by 15 exponentiations.

Remark: Recall that for the primes e_i that could be accumulated, the range restriction is $[A,\ B]$ with $2 < A$ and $B < A^2$. This restriction is mainly due to the fact that operations are achieved in $QR_{\widetilde{n}}$. The author of this thesis conjectures that by working in the group of τ power residues modulo $\widetilde{n}$, the upper bound B can be set to $B < A^{\tau}$, however the lower bound would be changed to $\tau < A$. Note that these groups are slightly different to *Schnorr groups* as the modular computations are performed with a special RSA modulus instead of a prime number. This conjecture should also be applicable in the case of [CL02a].

Chapter 4

Interactive Range Proofs

This chapter starts by presenting, in Section 4.1, the range proof primitive in its basic interactive version. Section 4.2 then continues with prior and recent work on interactive range proofs, as well as some related work. In Section 4.3, a family of range proofs based on the set membership proof primitive are presented. This will be achieved by using the Boneh-Boyen signature based set membership proof from Section 3.3. Section 4.4 introduces and explains the sumset representation of integer intervals. Based on the notion of sumsets, a more efficient range proof is presented in Section 4.5. The main results of Section 4.3 are published at Asiacrypt 2008 [CCs08], as a joint work with Jan Camenisch and abhi shelat[1]. The main theory and results of Section 4.4 and Section 4.5 are published in the proceedings of ACISP 2010 [CLs10], as a joint work with Helger Lipmaa and abhi shelat. Lastly, note that Protocol 4.4 in Section 4.3 is unpublished as it is a direct result of [CCs08].

4.1 Interactive Range Proofs Primitive

The problem tackled in this chapter is closely related to the set membership proof problem explained in the previous chapter. Indeed the *range proof* problem can be seen as a special case of the set membership proof problem, when the set Φ consists of all integers that are within a given range $[A,\ B]$. Hence $\Phi = \{x \in \mathbb{N} : A \leqslant x \leqslant B\}$, where $A, B \in \mathbb{N}$. For more clarity, we recall the game of the set membership proof problem and describe the range proof problem by way of a similar game between a *prover* and a *verifier*. In the range proof game, the prover wants to convince the verifier of the veracity of a specific statement. This statement is that his secret element σ that he picked (and fixed in a commitment available to any verifier) is included in the public range $[A,\ B]$, where $A, B \in \mathbb{N}$. This game comes with the same concerns as for the set membership proof problem. The prover wants to reveal no information besides the fact that his secret element belongs to the public range $[A,\ B]$, and that he is able to open

[1] Note that abhi shelat requires his name to be cited in lower case.

his commitment to such an element. As for the verifier, he wants to be sure that the prover is unable to cheat. Hence zero-knowledge and soundness properties are respectively needed to address these concerns. In this chapter, the focus is on the general interactive version of range proofs. Non-interactive range proofs will be the topic of the next chapter.

The need for *range proofs* started with the need to adapt cryptographic protocols constructed in idealized models, into protocols secure against any (malicious) adversary. Range proofs became even more necessary with the rise of electronic communications. As several online services became more and more complex, the need for complex cryptographic building boxes followed. Services such as *anonymous credentials*, *e-cash*, *e-auctions*, *electronic elections*, and *e-voting* are all examples of services that require range proof primitives. Hence range proofs are now considered a basic cryptographic building block. In the case of *anonymous credentials*, a typical example is age restriction services. Assume that a user needs to prove that her age is greater than 18 years to access some adult content, or between 13 and 18 in the case of teen-community websites; these can be ensured with range proofs performed on their hidden age contained in a passport credentials or electronic identities (or e-ID). These cases can be generalized to any timestamp credential that the owner wishes to keep secret. In the case of *e-cash* and *e-auctions*, range proofs become useful in providing range information on the size of portfolios or on the bid range. This can be illustrated with a user accessing a private investment platform or a fiscal arrangement platform (also called *lump sum taxation* in Switzerland). In these cases, a user needs to prove that the size of his portfolio is within some range in order to access these platforms. However, they also wish to keep their exact fortune a secret from these platforms. In the case of e-auctions, not only are users requested to prove that their portfolio is large enough to participate in the auctions, but for some specific auctions with sealed bids, such as blind auctions or Vickrey auction, bidders might need to prove that their bids are higher than a minimum threshold. Last but not least, *electronic elections* or *e-voting* with, respectively, a very large number of candidates or choices, can benefit from range proofs to attest the validity of ballots, as explained by Damgård and Jurik in [DJ01]. However, where the voting choices are small, such as in [CGS97] where there is only a choice between yes and no, the set membership proof primitive should be used instead of range proofs.

Definition 4.1 (Range Proof)
Let $C = (\text{Gen}, \text{Com}, \text{Open})$ *be the generation, the commit, and the open algorithm of a string commitment scheme. A range proof with respect to the commitment scheme* C *is a special case of the set membership proof in which the set* Φ *is a sequence of consecutive integers* $\Phi = [A, B]$ *for* $A, B \in \mathbb{N}$. *Hence, for an instance* c, *a range proof with respect to commitment scheme* C *and integer range* $[A, B]$ *is a proof of knowledge for the following statement:*

$$PK\left\{(\sigma, \rho) : c \leftarrow \text{Com}(\sigma; \rho) \wedge \sigma \in [A, B]\right\}, \text{ where } A, B \in \mathbb{N}.$$

Remark: As in the case of set membership proofs, the proof system for range proofs is defined for *any* commitment scheme. Moreover, the statement being proven is the ability of the prover

to open his commitment to an element contained in the public range $[A, B]$. Furthermore, it is important to note that interactive range proofs are, in fact, interactive arguments, for the exact same reasons as for set membership proofs. Since the cryptographic literature (past, present, and related) refers to the problem as a "range proof", that term is used in this thesis. Some additional explanations are provided in the remark in Section 3.1.

A naïve solution would be to use a set membership proof to solve the range proof problem. More efficient solutions can be obtained by exploiting the structure of Φ, as it is a consecutive integer range in the case of range proofs. Nevertheless, if the range is very small (under 7 elements), then it would be more efficient to directly employ the set membership proof protocol presented in Section 3.3.

Two honest verifier zero-knowledge solutions will be presented in this chapter, based on three moves protocols called Σ-protocols (see Section 2.4.1). The focus on honest verifiers is justified by the availability of the Cramer et al. transformation [CDM00] that converts honest verifier zero-knowledge proof systems so as to be secure against any verifier. Moreover, this transformation is well adapted for Σ-protocols. In these protocols, the prover sends an initial message containing some elements determining the randomness used to blind his secrets. The verifier then provides the prover with an unpredictable, random challenge, allowing the prover to reply with a final message in order to complete the proof system. Furthermore, the restriction of honest verifiers protocols facilitates comparisons with other range proofs, as the majority restrict themselves to this model.

The primary solution for range proofs used in this thesis is explained in Section 4.3. This solution is tightly linked to the set membership proof primitive, as it divides the integer range at hand into a u-base decomposition in order to obtain integer intervals that are small enough to be handled by a set membership proof protocol. The computational hardness assumptions required are identical to the ones needed for the set membership proof primitive. This solution offers two important improvements for range proofs. Initially applied using signature based set membership proofs, the technique of this solution was the first one to introduce the u-base decomposition of ranges and combine it with a proof of knowledge of a signature. The second improvement to range proofs is the asymptotical bound for the communication complexity, with respect to the honest verifier zero-knowledge security. Indeed, the communication complexity achieved by this solution is $O\left(\frac{k}{\log k - \log\log k}\right)$ group elements, with the security parameter $k = \log(B - A)$.

The second solution provides a constant factor 2 improvement on the communication complexity, compared to the primary solution. To achieve this improvement, the primary solution is modified so that it uses a *sumset* representation of the range instead of a specific u-base decomposition. Sumset representations of integer ranges will be explained in Section 4.4 before presenting the second solution in Section 4.5. Sumsets are classified in additive combinatorics as a multi-base decomposition. Furthermore, the computational hardness assumptions for this second solution are left unchanged from the primary solution.

4.2 Prior and Related Work

Known range proofs can be classified into four categories, according to their underlying techniques:

1. Σ***-response***[2] range testing ([BCDvdG87, CFT98a, CFT98b, FO98]);
2. ***positivity*** testing ([Bou00, Lip03, Gro05, Sce09]);
3. range ***decomposition***
   ([BG97, Mao98, DJ01, Sch01, LAN02, CCs08, CLs10, MN10, Gro11, CCJT13]); and
4. proof of ***signature*** knowledge[2] ([TS06, CCs08, CLs10]).

Historically speaking, the first solution to range proofs was created in 1987 by Brickell, Chaum, Damgård, and van de Graaf in [BCDvdG87]. Their solution was based on the discrete logarithm assumption and was achieved using a range check on the response message from a Σ-protocol. Therefore this technique is designated as a Σ***-response*** range testing. Unfortunately, too many drawbacks follow from this technique. In their protocol, a prover holding a secret element $\sigma \in [0, B]$ can only prove that $\sigma \in [-B, 2B]$ after repeating the proof in parallel k times, where k defines the soundness security of the protocol (the success probability of a malicious prover is upper bounded by 2^{1-k}). This inaccuracy in the range being proven is specific to this technique, and is measured with a factor called *expansion rate* δ. In the case of [BCDvdG87], the expansion rate is $\delta = 3$. The Σ-responses are often computed as $m = \sigma \cdot c + r$, where r is a random element that has been committed to at the beginning of the protocol and c is a random challenge provided by the prover. Checking solely the range of m cannot provide an accurate range proof with an expansion rate of $\delta = 1$, and at the same time ensuring the zero-knowledge property of the proof.

Following the lead of Brickell et al. [BCDvdG87], two other solutions were produced based on the Σ-response range testing. Using the same computational hardness assumption (namely the discrete logarithm assumption), Chan, Frankel, and Tsiounis overcame in [CFT98a, CFT98b] the need to repeat the proof k times. However, they ended up with a larger expansion rate of $\delta = 2^{2k+3}$ and a probabilistic completeness of $\left(1 - 2^{-k-1}\right)$ for a soundness security of 2^{-k}. This means that the completeness of their protocol will fail with probability 2^{-k-1} and a malicious prover will succeed with a probability of at most 2^{-k}. Moreover, for a secret element $\sigma \in [0, 2^H]$, the range statement being proven is $\sigma \in [-2^{H+2k+2}, 2^{H+2k+2}]$.
Although very similar to [CFT98a, CFT98b], Fujisaki and Okamoto used the strong RSA assumption in [FO98] (see Section 2.2.2). They achieved perfect completeness in computations performed with an RSA modulus N of size k, instead of computations modulo a prime p as in [BCDvdG87, CFT98a, CFT98b]. For a secret element $\sigma \in [A, B]$, the expansion rate that they obtained is $\delta = 2^{O(k)+1}$. Hence, for an RSA modulus N of 1024 bits, the expansion rate

[2] The use of this designation is specific to this thesis and is not mentioned elsewhere.

becomes $\delta \geqslant 2^{1025}$. Furthermore, their scheme is statistically witness indistinguishable (see Section 2.3.5) and not honest verifier zero-knowledge.

In order to limit the drawbacks of the Σ-response range testing, Boudot proposed, in [Bou00], to solve arbitrary range proofs $\sigma \in [A, B]$ with two ***positivity*** tests $B-\sigma \geqslant 0$ and $\sigma - A \geqslant 0$. In his solutions, the Fujisaki-Okamoto commitment scheme (see Section 2.4.2) is used to commit to σ. Each positivity test $m \geqslant 0$ is solved by finding the largest square $x^2 \leqslant m$. Thus, the positivity test is obtained by showing that $m = x^2 + x_\epsilon \geqslant 0$, where the commitment to x^2 is proven to contain a square using a group of unknown order, and $x_\epsilon \in \left[0,\ 2\sqrt{B-A}\right]$ is proven with the Chan et al. method [CFT98b]. As the latter method induces inaccuracy in the range proof, Boudot solves this issue by artificially increasing the secret with a positive constant 2^T, where $T = 2(2k+3) + (B-A)$. The positivity test becomes $m2^T = \tilde{x}^2 + \tilde{x}_\epsilon \geqslant 0$, where $\tilde{x}^2$ is the largest square $\tilde{x}^2 \leqslant m2^T$ and $\tilde{x}_\epsilon \in [0,\ 2\sqrt{2^T(B-A)}]$. Using Chan et al. method on $\tilde{x}_\epsilon$ will now convince the verifier that:

$$
\begin{aligned}
|\tilde{x}_\epsilon| &\leqslant \left(2\sqrt{2^T(B-A)}\right)\cdot 2^{2k+2} \\
&\leqslant \left(2^{1+T/2}\sqrt{B-A}\right)\cdot 2^{2k+2} \\
&\leqslant 2^{T/2+2k+3}\sqrt{B-A} \\
&\leqslant 2^{T/2}2^{2k+3}(B-A)^{1/2} \\
&< 2^{T/2}2^{(2k+3)}(2^{(B-A)})^{1/2}, \text{ as } B-A>0 \\
&< 2^{T/2}2^{(2k+3)}2^{(B-A)/2} \\
&< 2^{T/2}2^{(2k+3)+(B-A)/2} \\
&< 2^{T/2}2^{T/2} \\
&< 2^T.
\end{aligned}
$$

Note that here, $|\tilde{x}_\epsilon|$ is the absolute value of $\tilde{x}_\epsilon$. The verifier is thus convinced that m is of the form $m2^T = \hat{x}^2 + \hat{x}_\epsilon$, with $\hat{x}_\epsilon \in \left]-2^T,\ 2^T\right[$. This implies that m is of the form $m = \hat{x}^2 2^{-T} + \hat{x}_\epsilon 2^{-T}$. As m has to be an integer, $(\hat{x}_\epsilon 2^{-T}) \in \left]-1,\ 1\right[$, and $\hat{x}^2 \geqslant 0$, these imply that $m \geqslant 0$. Note also that the computational hardness assumption required in [Bou00] is the same as for [CFT98b], namely the strong RSA assumption. Furthermore, the interactive version of the Boudot range proof is a 7 round protocol, where 28 elements are transmitted for roughly 32'000 bits. Note that the number of elements transmitted is independent of the range size. This protocol therefore becomes advantageous for large ranges. Moreover, the verifier needs to compute 24 exponentiations, while the prover needs to compute 29 exponentiations.

In the category of positivity testing, Lipmaa recalled, in [Lip03], a Lagrange theorem from 1770, that stated that any positive integer m can be represented as the sum of four integer squares, $m = \sum_{i=1}^{4} \left(x_i^2\right)$. In order to compute these squares, Lipmaa provided an improved algorithm by combining the initial one proposed by Rabin and Shallit in [RS86] with an algebraic trick produced by Cornacchia in 1908 (and described in Section 1.5.2 of [Coh10]) to represent a prime p of the form $p = 1 \pmod 4$ as the sum of two squares. Hence, to solve the positivity

testing on an integer m, Lipmaa proves in [Lip03] that m is the sum of four integer squares. His protocol requires the strong RSA assumption, and a positivity test is achieved by transmitting 16 elements for roughly 18'000 bits. Hence both [Bou00] and [Lip03] are comparable in terms of communication complexity, if two positivity tests from [Lip03] are used to achieve the range proof $\sigma \in [A,B]$. Furthermore, the protocol presented in [Lip03] is a Σ-protocol, and as such requires only 3 rounds of communication. Moreover, the verifier needs to perform 18 exponentiations for one positivity test performed on an integer m. As for the prover, his computation complexity is dominated by 18 exponentiations and a polylogarithmic time complexity $O\big((\log m)^2\big)$ to find the four squares x_i^2, such that $m = \sum_{i=1}^{4} \left(x_i^2\right)$. Last but not least, the protocol by Lipmaa in [Lip03] has perfect completeness, which is a missing property in [Bou00].
Similarly to the Lagrange theorem, Legendre produced, in 1798, a theorem stating that any positive integer $m \neq 4^a(8b+7)$, for positive integers a and b, can be represented as the sum of three squares. Using this latter theorem instead of the Lagrange theorem, Groth proposed, in [Gro05], to use the exact same techniques as in [Lip03] with three squares instead of four. To compute the three squares, Groth uses an algorithm by Rabin and Shallit [RS86] with the Cornacchia enhancement, as it was used in the case of computing the four squares for [Lip03]. Hence, Groth reduces the problem of proving $m \geqslant 0$, to the problem of proving $4m+1 \geqslant 0$, as $4m+1$ can always be represented as the sum of three squares, due to the Legendre theorem. This saves the prover and the verifier from having to compute 4 exponentiations. Furthermore, the communication complexity is reduced to 14'720 bits, as 13 elements need to be transmitted independently of the range size, instead of 16 in the case of [Lip03].
Scemama suggested, in [Sce09], to solve general range proofs of the form $\sigma \in [A,B]$ by solving the positivity test $(B-\sigma)(\sigma - A) \geqslant 0$. In order to do so, he uses the positivity test of Boudot [Bou00], to prove that $(\sigma - A)(\sigma - B) = -(x^2 + x_e) \leqslant 0$. By doing so, Scemama obtains a 9 round protocol with probabilistic completeness. The communication complexity consists of 24 elements transmitted for roughly 28'500 bits. The computation complexity of the verifier is reduced to 21 exponentiations and that of the prover is reduced to 27 exponentiations.

A third approach to range proofs is to perform a ***range decomposition***. The idea is to decompose the secret element σ into some base, and then prove that the decomposition of σ is composed by elements of that base. The most trivial decomposition is the ***binary decomposition*** ([BG97, Mao98, DJ01, Sch01, LAN02, Sch05, MN10, Gro11, CCJT13]). More advanced techniques involve the use of ***u-ary decomposition*** ([CCs08]) and general ***multi-base decomposition*** ([CLs10]).

The ***binary decomposition*** *range proof* was introduced in 1997 by Bellare and Goldwasser in [BG97]. In order to prove that $\sigma \in [0,\ 2^k - 1]$, which means that σ is a k-bits string, they decompose σ in its binary form $\sigma = \sum_{i=0}^{k-1} \sigma_i 2^i$. Each σ_i is committed, then proven to be a binary element using a 1-out-of-2 elements proof of knowledge provided by Cramer, who privately disclosed it to Bellare and Goldwasser. The 1-out-of-2 elements proof of knowledge is obtained by applying the results of [CDS94] to the Schnorr protocol [Sch91]. This proof technique is referred to as an "OR-proof" in the current literature. It is then sufficient to prove

the correspondence between the σ_i commitments and the commitment to σ. The verifier is then convinced that the secret σ lies in $[0, 2^k - 1]$ since there were only k commitments. Moreover, the security of [BG97] relies on the discrete logarithmic assumption to provide perfect witness indistinguishability (see Section 2.3.5), a weaker security property than perfect zero-knowledge. The computational complexity for both the prover and the verifier is $O(k)$ exponentiations, while $O(k)$ group elements are transmitted. Note that the protocol presented in [Mao98] by Mao, is very similar to [BG97]. The sole improvement provided by Mao targets the correspondence between the commitment to σ and its binary decomposition commitments. Mao showed that this correspondence requires one exponentiation fewer when the randomnesses of the binary decomposition commitments sum up to the randomness of the commitment to σ.

Instead of the discrete logarithmic assumption, in [DJ01] Damgård and Jurik proposed the same structure as in [BG97, Mao98] while using the *decisional composite residuosity assumption* (DCR assumption) from Paillier [Pai99]. Informally, this assumption captures the difficulty of deciding if a random element $x \in_R \mathbb{Z}^*_{n^2}$ is a n'th power in $\mathbb{Z}^*_{n^2}$, where n is an RSA modulus. Here, x being a n'th power in $\mathbb{Z}^*_{n^2}$, implies that it can be written as $x = y^n \pmod{n^2}$, for a $y \in \mathbb{Z}^*_{n^2}$. For more details on the DCR assumption, see Section 2.2.2. The protocol proposed in [DJ01] uses computations modulo n^{s+1}, for any $s \geqslant 1$. Unfortunately, the asymptotical communication and computational complexities of this latter protocol are unchanged when compared to the previous range proofs using the binary decomposition method.

Schoenmakers in [Sch01, Sch05] studied and discussed how to solve a more general case $\sigma \in [0, B]$ where $2^{k-1} < B \leqslant 2^k$, from the binary decomposition of [BG97, Mao98]. His method consists of achieving the range proof $\sigma \in [0, B]$ with either a conjunction or a disjunction of two binary decomposition range proofs:

$$\sigma \in [0, B] \iff \sigma \in [0, 2^k] \ \wedge \ \sigma \in [B - 2^k, B]$$
$$\sigma \in [0, B] \iff \sigma \in [0, 2^{k-1}] \ \vee \ \sigma \in [B - 2^{k-1}, B].$$

He also introduced several recursive relations which can be used to reduce the number of basic proofs of knowledge required when committing to the individual bits of the secret. More precisely, he writes the upper bound B of the positive range $[0, B]$ as either the product or the sum of two numbers. By performing this scheme recursively, he decreased the amount of work needed. However, the overall communication complexity still consists of $O(k)$ transmitted elements for a computational load of $O(k)$ exponentiations. Nevertheless, Schoenmakers noticed that for $k \leqslant 27$, the binary decomposition range proof is more efficient with regards to the communication complexity, when compared to the positivity test from Groth method [Gro05]. Note also that the techniques of Schoenmakers for reducing certain ranges to other more convenient ranges can be used with any range proof technique.

Lipmaa, Asokan, and Niemi explained, in [LAN02], an interesting method for the range proof $\sigma \in [0, B]$, based on a binary sumset representation. As the sumset representation will be explained in more details in Section 4.4, only the binary case is stated here. The secret element is decomposed as $\sigma = \sum_{i=0}^{\lfloor \log B \rfloor} \sigma_i B^i$, where $\sigma_i \in \{0, 1\}$ and $B_j = \left\lfloor \frac{B + 2^i}{2^{i+1}} \right\rfloor$. This can be seen as a

generalization of the method introduced in [DJ01], which works only where $B = 2^k - 1$. Hence the same asymptotical communication and computational load are achieved for the same computational hardness assumptions.
An alternative protocol based on the discrete logarithmic assumption has been proposed by Moran and Naor in [MN10], where they substitute the k proofs of knowledge that the elements σ_i are binary, with a proof that the set $(\sigma_i,\ 1-\sigma_i) : \sigma = \sum_{i=0}^{k-1} \sigma_i 2^i$ is a shuffle of the set $\{C_0,\ C_1\}^k$, where C_0 and C_1 are respectively commitments to 0 and 1. Although the idea seems more elegant, the burden of the shuffle argument increases the computational complexity to $O(k^2)$ exponentiations and requires $O(k^2)$ group elements to be transmitted.
Groth subsequently proposed another solution based on the binary decomposition method in [Gro11]. By using a rather complicated method based on commitments of commitments, he claims to achieve a range proof with communication complexity of $O(k^{1/3})$ group elements. His protocol relies on the common reference string model (see Section 2.2.4) and requires 7 rounds of communication. Furthermore, the computational hardness assumption needed is the reverse double pairing assumption in asymmetric bilinear groups (for more details see Section 2.2.2). As for the security proof, soundness is achieved by the use of a witness-extended emulation (see Section 2.3.3). When studying this protocol, special care should be taken, as some small inaccuracies are present. For instance, the range proof uses a batch proof argument in which the elements c_{u_j} and c_{v_j} should be sent in step 5 instead of step 3. Last but not least, the computational complexity is higher than claimed. A prover will need to compute $O(k^{2/3})$ exponentiations and $O(k^{2/3})$ pairings, while a verifier will need to compute $O(k^{1/3})$ exponentiations and $O(k^{1/3})$ pairings. These inaccuracies were notified to the author, who provided us with a private corrected version. A public corrected version should soon be published.
More recently, Canard et al. presented, in [CCJT13], a general range proof $\sigma \in [A, B]$ inspired by the binary sumset representation of Lipmaa et al. [LAN02] with the Fischlin lemma on binary representations, as described in [Fis01]. Informally, the Fischlin lemma states that for any type of binary representation, when comparing representations of two different elements, the higher order bits are identical. Furthermore, the first occurrence of a difference reveals a 0 bit for the smaller element and a 1 bit for the larger element. Moreover, the range proof protocol is provided without any security proofs. It should be further noted that although their protocol claims to use a multi-base decomposition, it is in fact a simple binary sumset representation. Their protocol could therefore be enhanced by using the general sumset representation. Moreover, they claim that their range proof protocol is only interesting for ranges that are smaller than $2^5 = 32$, which would lead to a communication load of at least 30 group elements (requiring the DDH assumption). For this kind of restriction, set membership proof protocols are more efficient, as shown in Figure 3.1.

The ***u-ary decomposition*** for range proofs was introduced by Camenisch, Chaabouni, and shelat in [CCs08] and will be discussed in Section 4.3. Furthermore, this method was enhanced by Chaabouni, Lipmaa, and shelat, in [CLs10], by applying general sumset representations of ranges, and is regarded as a ***multi-base decomposition***. This will be detailed in Section 4.5.

Regarding ***signature*** based range proofs, Teranishi and Sako proposed, in [TS06], to simply apply their signature based set membership proof primitive for the range proof. Camenisch et al. in [CCs08], followed by Chaabouni et al. in [CLs10] provide a much more efficient solution, by combining the base decomposition method with proofs of signature knowledge.

It should be noted that in comparison with the notion of range proofs that we defined earlier, the protocols presented by Nergiz et al. in [NNPC10] and Wu et al. in [WHLD14], are slightly different as the statement $\sigma \in [\alpha, \beta]$ being proven should also hide the range $[\alpha, \beta]$ in which the membership is proven.

Last but not least, a comparison between relevant protocols, based on a concrete example, is provided in Figure 4.6 at the end of this chapter.

4.3 Set Membership Based Range Proofs

In this section, the general range proof problem $\sigma \in [A, B]$ is reduced to solving the range proof problem $\sigma \in [0, u^\ell)$. Furthermore, this latter problem is solved by decomposing the range in the u-ary base, for some optimally chosen u. Thus, each element σ of the range $[0, u^\ell)$ can be identified with ℓ elements $\sigma_j \in [0, u-1]$ such that $\sigma = \sum_{j=0}^{\ell-1} \sigma_j u^j$. Hence, in order to show that a commitment holds a secret element $\sigma \in [0, u^\ell)$, it suffices to show that its decomposition in the u-ary base leads to ℓ commitments of elements in the range $[0, u-1]$. Therefore, the key technique is to use a set membership proof protocol in order to prove that each committed digit σ_j is indeed a digit in base u. Note that in the case of $u = 2$, this becomes a simple binary decomposition. Writing the secret in base-u (instead of base 2) is indeed an obvious step. However, using prior methods, doing so does not reduce the communication complexity, nor the computational complexity. Using prior methods, proving that a committed digit is a u-ary digit requires $(u-1)$ OR-proofs, forcing complexities to be linear in the security parameter. By using the set membership proof primitives introduced in the previous chapter, the complexities can be reduced both asymptotically as well as in practice for many frequently occurring ranges. This will be explained using the particular example of the Boneh-Boyen signature based set membership proof from Section 3.3.

The key insight is the design of a scheme that can reuse the elements from the u-ary base proof, in all ℓ proof instances. Specifically, the verifier can send *one* list of u signatures representing the u-ary digits, and the prover can use this *same* list to prove that all ℓ digits are indeed u-ary digits. Thus, the total communication complexity of our approach is $O(u+\ell)$. With appropriately selected values for u and ℓ, we show that this approach yields a proof of size $O\left(\frac{k}{\log k - \log\log k}\right)$ group elements, where k is the security parameter. Compared to previous literature results, this leads to better asymptotical and practical complexities for both communication and computation loads.

Note, however, that if the range is small or the same range is used for many protocols, then it is

more efficient to employ the set membership proof protocol directly. This choice is implicitly contained in the choice of u and ℓ. Indeed, for $\ell = 1$, the range proof presented in this section is identical to the set membership proof on which it is initially based.

Computational assumptions. The protocol in this section depends solely on the choice of the set membership proof primitive and on the commitment scheme used. These choices will dictate the necessary computational hardness assumptions. For the Boneh-Boyen signature based set membership proof in conjunction with Pedersen commitments, the range proof requires bilinear groups (see Section 2.1.3), associated computational hardness assumptions as well as the q-Strong Diffie Hellman assumption (see Section 2.2.2). The computational hardness assumptions relating to bilinear groups, as well as bilinear groups altogether, can be relaxed if the protocol employs the techniques from Canard et al. in [CCJT13] in order to compute Boneh-Boyen signatures without pairings. This matter will be discussed at the end of this section, as the focus will be on the original version presented in [CCs08].

Let PG be a pairing[3] group generator that on input 1^κ outputs descriptions of multiplicative groups $\mathbb{G}_1$ and $\mathbb{G}_T$ of prime order p where $\|p\| = \kappa$. Let $\mathbb{G}_1^* = \mathbb{G}_1 \setminus \{1\}$ and let $g \in \mathbb{G}_1^*$. Let e be the corresponding admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$. Recall that for all $a, b \in \mathbb{Z}_p$ it holds that $e(g^a, g^b) = e(g, g)^{ab}$, that $e(g, g) \neq 1$, and that e is efficiently computable.

Protocol explanation. The first range proof presented here in Protocol 4.1 is based on the set membership proof from Section 3.3. The common input includes the following elements: a description of $\mathbb{G}_1$ and $\mathbb{G}_T$, as provided by the pairing group generator PG for the Boneh-Boyen signature; two generators g and h of $\mathbb{G}_1$ for the Pedersen commitment; the parameters u and ℓ defining the public range $[0, u^\ell)$ such that $u^\ell < |\mathbb{G}_1|$; and a Pedersen commitment C to one element in $[0, u^\ell)$. As $|\mathbb{G}_1| = p$, this implies that $\mathbb{Z}_u \subset \mathbb{Z}_p$. The prover input additionally contains elements σ and r such that $C = g^\sigma h^r$ and $\sigma \in [0, u^\ell)$.

Notice that the first message exchanged is identical to the first message of the Boneh-Boyen signature based set membership proof protocol (see Protocol 3.1 in the previous chapter), where $\Phi = \mathbb{Z}_u$. Hence this first message consists of the Boneh-Boyen public key y of the verifier together with signatures A_i on every element contained in $\mathbb{Z}_u$. Here, $x \in \mathbb{Z}_p^*$ is the Boneh-Boyen secret key of the verifier. As an alternative, a trusted third party can be employed to produce the signatures on every element of $\mathbb{Z}_u$. Moreover, picking $x \in \mathbb{Z}_p^*$ should be done such that $-x$ is not present in $\mathbb{Z}_u$. It would not be possible for the honest verifier to produce a signature on the element $-x$, as he would need to perform an inversion of 0 modulo p. As for the malicious verifier, it would not be possible to provide a signature A_i such that $e(g, g) = e(A_i,\ y \cdot g^i)$ for $y = g^{-i}$. In all cases, the correctness of the public key y and signatures A_i should be checked by the prover (and by the verifier if a trusted third party generated them).

[3] Note that here, the use of symmetric bilinear groups is to ease explanations. In practice, asymmetric bilinear groups could and should be used as explained in Section 2.1.3.

For the second message, the prover decomposes his secret σ in base u to obtain the u-ary digits $\{\sigma_j\}$, such that $\sigma = \sum_{j=0}^{\ell-1}(\sigma_j u^j)$. For each u-ary digit σ_j, the prover selects a random secret v_j and composes a blinding V_j on the signature A_{σ_j} of σ_j. This blinding is achieved by the exponentiation $(A_{\sigma_j})^{v_j}$. The second message is then the collection of the blindings: $\{V_j\}$, for $j \in \mathbb{Z}_\ell$. Recall that the protocol presented here is performed in the honest verifier model. The standard checks, such as verifying that an element is in the correct group, are mainly provided for the sake of completeness. However, they become necessary when compiling from the honest verifier zero-knowledge model to the full zero-knowledge proofs.

Common Input: g, h, u, ℓ, and a commitment C.
Prover Input: σ, r such that $C = g^\sigma h^r$ and $\sigma \in [0, u^\ell)$.

$P \xleftarrow{y,\{A_i\}} V$
- Verifier picks $x \in_R \mathbb{Z}_p^*$ such that $-x \notin \mathbb{Z}_u$ and sends $y \leftarrow g^x$ and $A_i \leftarrow g^{\frac{1}{x+i}}$, $\forall i \in \mathbb{Z}_u$.
- Prover checks that $y \in \mathbb{G}_1^*$, $A_i \in \mathbb{G}_1^*$ and that $e(g,g) \stackrel{?}{=} e(A_i,\ y \cdot g^i)$, $\forall i \in \mathbb{Z}_u$.

$P \xrightarrow{\{V_j\}} V$
- Prover picks $v_j \in_R \mathbb{Z}_p^*$ and sends $V_j \leftarrow A_{\sigma_j}^{v_j}$, for every $j \in \mathbb{Z}_\ell$, such that $\sigma = \sum_{j=0}^{\ell-1}(\sigma_j u^j)$.
- Verifier checks that $V_j \in \mathbb{G}_1^*$, $\forall j \in \mathbb{Z}_\ell$.

Prover and Verifier run

$$\mathrm{PK}\left\{\left(\{\sigma_j\}, r, \{v_j\}\right):\ C = h^r g^{\sum_{j=0}^{\ell-1}(\sigma_j u^j)} \ \wedge\ V_j = g^{\frac{v_j}{x+\sigma_j}},\ \forall j \in \mathbb{Z}_\ell\right\}$$

$P \xrightarrow{\{a_j\},D} V$
- Prover picks $s_j, t_j, m \in_R \mathbb{Z}_p^*$ for every $j \in \mathbb{Z}_\ell$ and sends $a_j \leftarrow e\left(V_j^{-s_j} g^{t_j}, g\right)$ and $D \leftarrow h^m g^{\sum_{j=0}^{\ell-1}(s_j u^j)}$.
- Verifier checks that $a_j \in \mathbb{G}_T$, $\forall j \in \mathbb{Z}_\ell$, and $D \in \mathbb{G}_1^*$.

$P \xleftarrow{c} V$
- Verifier sends a random challenge $c \in_R \mathbb{Z}_p^*$.
- Prover checks that $c \in \mathbb{Z}_p^*$.

$P \xrightarrow{\{z_{\sigma_j}\},\{z_{v_j}\},z_r} V$
- Prover sends $z_r \leftarrow (m - rc)$, and $z_{\sigma_j} \leftarrow (s_j - \sigma_j c)$, $z_{v_j} \leftarrow (t_j - v_j c)$ for every $j \in \mathbb{Z}_\ell$.
- Verifier checks that $z_{\sigma_j}, z_{v_j}, z_r \in \mathbb{Z}_p^*$ for every $j \in \mathbb{Z}_\ell$, that $D \stackrel{?}{=} C^c h^{z_r} g^{\sum_{j=0}^{\ell-1}\left(z_{\sigma_j} u^j\right)}$ and that $a_j \stackrel{?}{=} e\left(V_j, y\right)^c \cdot e\left(V_j^{-z_{\sigma_j}} g^{z_{v_j}}, g\right)$ for every $j \in \mathbb{Z}_\ell$.

Protocol 4.1 – Interactive range proof protocol for range $[0, u^\ell)$

After the blindings of the u-ary digit signatures have been transmitted, the prover and verifier engage in a proof of knowledge that the blinded signatures V_j correspond to the secret u-ary

digits σ_j, and that these digits are the u-ary decomposition of the secret σ contained in the initial commitment. To do so, the prover selects the random parameters s_j, t_j, and m in $\mathbb{Z}_p^*$ that will be used to blind his secret elements σ_j, v_j, and r respectively. The initial message of the proof of knowledge is then a commitment D on the secret random elements s_j used by the prover, and the group elements $a_j \in \mathbb{G}_T$ which are bilinear pairings on the blinded signature V_j using the corresponding secret randomness s_j contained in the commitment D. At the end, a_j will allow the verifier to check that V_j contains a valid blinded signature. Note as well that $D \in \mathbb{G}_1^*$. If $D = 1$, the prover would be able to retrieve the discrete logarithm of h in base g by outputting:

$$\log_g h = \frac{-\sum_{j=0}^{\ell-1}\left(s_j u^j\right)}{m} \pmod{p}.$$

Following this step, the verifier challenges the prover with a single challenge c. After receiving this challenge, the prover replies with the elements z_{σ_j}, z_{v_j}, and z_r. These will allow the verifier to complete the proof of knowledge by verifying that the following equations hold:

$$D \stackrel{?}{=} C^c h^{z_r} g^{\sum_{j=0}^{\ell-1}\left(z_{\sigma_j} u^j\right)}$$
$$a_j \stackrel{?}{=} e\left(V_j, y\right)^c \cdot e\left(V_j^{-z_{\sigma_j}} g^{z_{v_j}}, g\right), \quad \forall j \in \mathbb{Z}_\ell.$$

Theorem 4.1

If the u-Strong Diffie-Hellman assumption associated with a pairing generator PG *holds, then Protocol 4.1 is a zero-knowledge range argument for the range* $[0, u^\ell)$.

Proof

Recall that $(\mathbb{G}_1, \mathbb{G}_T) \leftarrow \mathsf{PG}(1^k)$, $p = |\mathbb{G}_1|$ and that $u^\ell < p < 2^k$. Similarly to the case of the Boneh-Boyen signature based set membership proof primitive, u is much smaller than p. The attack from Cheon [Che06] on the u-Strong Diffie-Hellman assumption states that the computational complexity of recovering the secret key x is of $O\left(\sqrt{p/u}\right)$ group operations, instead of $O\left(\sqrt{p}\right)$ group operations. Hence the computational complexity reduction of $O\left(\sqrt{u}\right)$ is, in our case, polynomially bounded. Furthermore, in order for the computational complexity of recovering the secret key to be higher than 2^κ group operations (κ-bit security), the difference between u and p should be higher than 2κ bits. This implies that for a κ-bit security, the security parameter k should be $k \geqslant 2\kappa + \log_2 u$.

To show that Protocol 4.1 is a zero-knowledge range argument, three security properties need to be met: the *completeness* of the protocol, the *special soundness* property, and the *special honest verifier zero-knowledge* property.

The *completeness* of the protocol follows by inspection. In particular, the last equalities hold

as follows. Recall that $D = h^m g^{\sum_{j=0}^{\ell-1}(s_j u^j)}$. Hence,

$$\begin{aligned} C^c h^{z_r} g^{\sum_{j=0}^{\ell-1}\left(z_{\sigma_j} u^j\right)} &= C^c \cdot h^{m-rc} \cdot g^{\sum_{j=0}^{\ell-1}((s_j-\sigma_j c)u^j)} \\ &= C^c \cdot h^m h^{-rc} \cdot g^{\sum_{j=0}^{\ell-1}(s_j u^j)} g^{-\sum_{j=0}^{\ell-1}(\sigma_j c u^j)} \\ &= C^c \cdot \left(h^r g^{\sum_{j=0}^{\ell-1}(\sigma_j u^j)}\right)^{-c} \cdot h^m g^{\sum_{j=0}^{\ell-1}(s_j u^j)} \\ &= h^m g^{\sum_{j=0}^{\ell-1}(s_j u^j)} \\ &= D. \end{aligned}$$

Similarly, as $a_j = e\left(V_j^{-s_j} g^{t_j}, g\right)$, the following holds:

$$\begin{aligned} e(V_j, y)^c \cdot e\left(V_j^{-z_{\sigma_j}} g^{z_{v_j}}, g\right) &= e(V_j, y)^c \cdot e(V_j, g)^{-z_{\sigma_j}} \cdot e(g, g)^{z_{v_j}} \\ &= e(V_j, g^x)^c \cdot\ e(V_j, g)^{-s_j + \sigma_j c} \cdot\ e(g, g)^{t_j - v_j c} \\ &= e(V_j, g)^{xc} \cdot\ e(V_j, g)^{-s_j} \cdot\ e(V_j, g)^{\sigma_j c} \cdot\ e(g, g)^{t_j} \cdot\ e(g, g)^{-v_j c} \\ &= e(V_j, g)^{xc + \sigma_j c} \cdot\ e(g, g)^{-v_j c} \cdot\ e(V_j, g)^{-s_j} \cdot\ e(g, g)^{t_j} \\ &= e(g^{\frac{v_j}{x+\sigma_j}}, g)^{(x+\sigma_j)c} \cdot\ e(g, g)^{-v_j c} \cdot\ e(V_j, g)^{-s_j} \cdot\ e(g, g)^{t_j} \\ &= e(g^{v_j}, g)^c \cdot\ e(g, g)^{-v_j c} \cdot\ e(V_j, g)^{-s_j} \cdot\ e(g, g)^{t_j} \\ &= e(g, g)^{v_j c} \cdot\ e(g, g)^{-v_j c} \cdot\ e(V_j, g)^{-s_j} \cdot\ e(g, g)^{t_j} \\ &= e(V_j, g)^{-s_j} \cdot\ e(g, g)^{t_j} \\ &= e\left(V_j^{-s_j} g^{t_j}, g\right) \\ &= a. \end{aligned}$$

The *special soundness* follows from the extraction property of the proof of knowledge and the unforgeability of the Boneh-Boyen signature scheme (Lemma 3.1). The extraction property is almost identical to the one related to the Boneh-Boyen signature based set membership proof protocol (for more details see the proof of Theorem 3.2). The difference appears in the witness that is output by the extractor. In the set membership proof protocol, the extractor outputs a witness (σ, r, v), whereas in this range argument, the witness will be $(\sigma, r, \{v_j\})$. The computations performed by the extractor will be explained first. A demonstration will also be provided, showing that if a malicious prover P^* is able to convince a verifier, then the extractor interacting with this prover P^* can either be used to break the unforgeability property of the Boneh-Boyen signature, or to create $\sigma \in [0, u^\ell)$ and r such that $C = g^\sigma h^r$.

At first, the extractor obtains two related transcripts tr and tr', for different challenges $c \neq c'$ but with the same initial elements $\{y, \{A_i\}, \{V_j\}, \{a_j\}, D\}$:

$$\begin{aligned} tr &= \{y, \{A_i\}, \{V_j\}, \{a_j\}, D, c, \{z_{\sigma_j}\}, \{z_{v_j}\}, z_r\} \\ tr' &= \left\{y, \{A_i\}, \{V_j\}, \{a_j\}, D, c', \{z'_{\sigma_j}\}, \{z'_{v_j}\}, z'_r\right\}. \end{aligned}$$

Then, the witness $(\sigma, r, \{v_j\})$ can be obtained by computing:

$$\sigma = \frac{\sum_{j=0}^{\ell-1}\left(z_{\sigma_j} - z'_{\sigma_j}\right)u^j}{c'-c}; \quad r = \frac{z_r - z'_r}{c'-c}; \quad v_j = \frac{z_{v_j} - z'_{v_j}}{c'-c}.$$

Notice that for the same reasons as in the proof of Theorem 3.2, $v_j \neq 0$, for every $j \in \mathbb{Z}_\ell$. Moreover, the extractor succeeds since $(c'-c)$ is invertible in $\mathbb{Z}_p$.

In the case that a malicious prover P^* convinces a verifier V, then P^* can be (almost) directly used to mount a weak chosen-message attack against the Boneh-Boyen signature scheme exactly as in the case of the Boneh-Boyen signature based set membership proof protocol. In the beginning, the attacker learns all of the signatures of the elements in $\mathbb{Z}_u$. As P^* succeeds in convincing V, the extractor will output the witness $(\sigma, r, \{v_j\})$, with $V_j = g^{\frac{v_j}{x+\sigma_j}}$ and $C = g^\sigma h^r$, where $\sigma = \sum_{j=0}^{\ell-1}(\sigma_j u^j)$. Hence, as $v_j \neq 0$ then $V_j^{(1/v_j)}$ is a valid signature of σ_j. Due to the unforgeability property of the Boneh-Boyen signature scheme, the extractor outputs $\sigma \in [0, u^\ell)$ and r such that $C = g^\sigma h^r$.

---

1. Sim retrieves $y, \{A_i\}$ from V^* (or from a trusted third party).
2. Sim chooses $\sigma \in_R [0, u^\ell)$, $v_j \in_R \mathbb{Z}_p^*$ for every $j \in \mathbb{Z}_\ell$, and computes $V_j \leftarrow A_{\sigma_j}^{v_j}$ where $\sigma = \sum_{j=0}^{\ell-1}(\sigma_j u^j)$.
3. Sim runs the simulator of
$$\mathrm{PK}\left\{\left(\{\sigma_j\}, r, \{v_j\}\right) : C = h^r g^{\sum_{j=0}^{\ell-1}(\sigma_j u^j)} \ \wedge \ V_j = g^{\frac{v_j}{x+\sigma_j}}, \ \forall j \in \mathbb{Z}_\ell\right\}:$$
   (a) On challenge $c \in \mathbb{Z}_p^*$, Sim chooses $\{z_{\sigma_j}\}, \{z_{v_j}\}, z_r \in_R \mathbb{Z}_p^*$.
   (b) Finally, Sim computes $D \leftarrow C^c h^{z_r} g^{\sum_{j=0}^{\ell-1}\left(z_{\sigma_j} u^j\right)}$ and $a_j \leftarrow e(V_j, g)^{-z_{\sigma_j} - \sigma_j c} e(g,g)^{z_{v_j} + v_j c}$ for every $j \in \mathbb{Z}_\ell$.
4. Sim returns the transcript $\left\{y, \{A_i\}, \{V_j\}, \{a_j\}, D, c, \{z_{\sigma_j}\}, \{z_{v_j}\}, z_r\right\}$.

Figure 4.1 – Simulator for the interactive range argument protocol

The *special honest verifier zero-knowledge* property follows from the perfect blinding of the signatures in the first phase, and the corresponding special honest verifier zero-knowledge property of the underlying proof of knowledge. Moreover, the interactions between any verifier V^* and any honest prover P can be efficiently simulated with the help of the simulator Sim, depicted in Figure 3.2. The simulator Sim will first follow the initialization and the blinding instructions honestly, using a random $\sigma \in_R [0, u^\ell)$ and random $v_j \in_R \mathbb{Z}_p^*$ to compute every V_j. Then Sim runs the simulator of the Σ-protocol for the underlying proof of knowledge:

$$\mathrm{PK}\left\{\left(\{\sigma_j\}, r, \{v_j\}\right) : C = h^r g^{\sum_{j=0}^{\ell-1}(\sigma_j u^j)} \ \bigwedge \ V_j = g^{\frac{v_j}{x+\sigma_j}}, \ \forall j \in \mathbb{Z}_\ell\right\}.$$

On the challenge $c \in \mathbb{Z}_p^*$ and for every $j \in \mathbb{Z}_\ell$, the simulator first randomly picks z_{σ_j}, z_{v_j}, z_r in $\mathbb{Z}_p^*$, then computes a_j and D as follows:

$$\begin{aligned} a &= e(V_j, g)^{-z_{\sigma_j} - \sigma_j c} e(g, g)^{z_{v_j} + v_j c}, \\ D &= C^c h^{z_r} g^{\sum_{j=0}^{\ell-1} \left(z_{\sigma_j} u^j \right)}. \end{aligned}$$

The output of the simulator is a transcript $\left\{ y, \{A_i\}, \{V_j\}, \{a_j\}, D, c, \{z_{\sigma_j}\}, \{z_{v_j}\}, z_r \right\}$, which has an identical probability distribution to a normal transcript between regular provers and verifiers. We can easily see that y, $\{A_i\}$, and c are identical in both transcripts, as they are provided by the verifier. The elements V_j have the same probability distribution as they are computed from a valid $\sigma \in [0, u^\ell)$ and random $v_j \in_R \mathbb{Z}_p^*$. As s_j, t_j, and m are randomly picked in $\mathbb{Z}_p^*$, they respectively impose the same randomness towards z_{σ_j}, z_{v_j}, and z_r. Hence $z_{\sigma_j}, z_{v_j}, z_r$ have the same probability distribution as well. It is straightforward to see that D has the same probability distribution as it is computed from the same elements with the same distributions. Last but not least, as $s_j = z_{\sigma_j} + \sigma_j c$ and $t_j = z_{v_j} + v_j c$, a_j has the same probability distribution for the same reasons as for D. Last but not least, since $\mathbb{G}_1$ is a prime-order group, the blinding is perfect in the first two steps of the simulator; thus the zero-knowledge property follows from the zero-knowledge property of the Σ-protocol in the third step. ■

Communication and Computational Complexity. As the first message of Protocol 4.1 should be regarded as being part of the setup procedure, it will not be included in the complexity analysis. Nevertheless, its cost is mentioned for the general purposes of comparison and analysis. This message consists of u signatures and the Boneh-Boyen public key y. These sum up to $(u+1)$ group elements of $\mathbb{G}_1$ for the communication, $(u+1)$ exponentiations in $\mathbb{G}_1$ for the verifier (or the trusted third party), and $(u+1)$ bilinear pairings for the prover in the non-honest verifier model.

Protocol 4.1 can be analyzed in two phases. the first phase consists of the prover sending ℓ blinded signatures V_j, which sum up to ℓ group elements in $\mathbb{G}_1$. The second phase consists of the underlying proof of knowledge, with 3 messages exchanged. The prover sends ℓ group elements in $\mathbb{G}_\mathrm{T}$ and one group element in $\mathbb{G}_1$, to which the verifier replies with a single challenge element in $\mathbb{Z}_p^*$. The last message, sent by the prover, consists of $(2\ell+1)$ elements from the group $\mathbb{Z}_p$. Overall, if $\|\mathbb{G}_1\|$, $\|\mathbb{G}_\mathrm{T}\|$, and $\left\|\mathbb{Z}_p\right\|$ correspond, respectively, to the binary size of the group elements in $\mathbb{G}_1$, $\mathbb{G}_\mathrm{T}$, and $\mathbb{Z}_p$, then the overall communication load Com according to parameters u and ℓ becomes:

$$Com(u, \ell) = \ell \cdot \left(\|\mathbb{G}_1\| + \|\mathbb{G}_\mathrm{T}\| + 2 \cdot \left\|\mathbb{Z}_p\right\| \right) + \left(\|\mathbb{G}_1\| + 2 \cdot \left\|\mathbb{Z}_p\right\| \right).$$

Note that the choice of u and ℓ are correlated with the size of the range. Hence a range proof for the range $[0, B)$ imposes the restriction $u^\ell \geqslant B$, where u and ℓ are chosen to be as small as possible. This restriction implies that $B \geqslant u \geqslant 2$. For the purpose of comparison,

assume that the size of the range is $(k-1)$ bits, in order to comply with the restrictions of the protocol: $B \leqslant u^{\ell} < |\mathbb{G}_1| < 2^k$. The choice $u = 2$ and $\ell = (k-1)$ corresponds to the settings of the binary decomposition protocol, which leads to a total communication complexity of $O(k)$ group elements. As for the other limit, the choice $u = B$ and $\ell = 1$ corresponds to the signature based set membership proof. Although this would lead to a constant communication complexity, with 7 group elements being transmitted, the cost of the setup phase would become exponential in k with the need to compute and transmit $O(2^k)$ signatures.

A more suitable choice of u is afforded by Protocol 4.1, as it allows more flexibility in the values of u. This can be demonstrated with the following choice for u:

$$\begin{aligned} u &= \frac{\log B}{\log\log B} = \frac{k-1}{\log(k-1)} \\ &\Longrightarrow u = O\left(\frac{k}{\log k}\right). \end{aligned} \tag{4.1}$$

The restriction $u^{\ell} \geqslant B$ implies that $\ell \geqslant \frac{\log B}{\log u} = \frac{k-1}{\log u}$. Furthermore, when taking into account equation (4.1), this restriction becomes:

$$\begin{aligned} \ell \geqslant \frac{k-1}{\log u} &\Longrightarrow \ell \geqslant \frac{k-1}{\log(k-1) - \log\log(k-1)} \\ &\Longrightarrow \ell = O\left(\frac{k}{\log k - \log\log k}\right). \end{aligned}$$

This results in a total communication complexity of:

$$Com(u,\ell) = O\left(\frac{k}{\log k - \log\log k}\right),$$

which is asymptotically smaller than $O(k)$.

Furthermore, not only is Protocol 4.1 asymptotically better, it also performs well for realistic concrete parameters. In that regard, the concrete optimization needs to take into account the setup cost. This leads to a general communication load of:

$$GenCom(u,\ell) = c_1 u + c_2 \ell + c_3, \tag{4.2}$$

where c_1, c_2, and c_3 are constants such that:

$$\begin{aligned} c_1 &= \|\mathbb{G}_1\|, \\ c_2 &= \|\mathbb{G}_1\| + \|\mathbb{G}_T\| + 2 \cdot \left\|\mathbb{Z}_p\right\|, \text{ and} \\ c_3 &= 2 \cdot \|\mathbb{G}_1\| + 2 \cdot \left\|\mathbb{Z}_p\right\|. \end{aligned}$$

By including in equation (4.2) the approximation $\ell \approx \frac{\log B}{\log u}$, the general communication load

becomes:

$$GenCom(u,\ell) = c_1 u + \frac{c_2 \log B}{\log u} + c_3. \tag{4.3}$$

A minimum can thus be found with respect to u, by setting the derivative of equation (4.3) to 0, and by attempting to solve the resulting equation:

$$\frac{d}{du} GenCom(u,\ell) = c_1 - \frac{c_2 \log B}{u(\log u)^2} = 0,$$

which simplifies to:

$$u(\log u)^2 = \frac{c_2}{c_1} \log B. \tag{4.4}$$

Unfortunately, equation (4.4) cannot be solved analytically. However, given c_1, c_2 and B, the numerical method of Kelly Black presented in [Bla97] can be used to find u.

Last but not least, regarding computational complexity, the honest verifier setting is assumed. Hence, the prover computational cost is dominated by $(3\ell+2)$ exponentiations and ℓ pairings. The verifier computational cost is dominated by $(2\ell+4)$ exponentiations and 2ℓ pairings.

Handling Arbitrary Ranges. Protocol 4.1 works for the range $[0, u^\ell)$. In order to handle an arbitrary range $[A,B]$, an improvement to the folklore reduction described by Schoenmakers in [Sch01] and [Sch05] can be used. First, it is straightforward to see that $[A,B] = [A,B+1)$. To achieve minimum communication complexity, the criteria for u in equation (4.4) becomes

$$u(\log u)^2 = \frac{c_2}{c_1} \log(B+1-A). \tag{4.5}$$

Suppose, initially, that $B+1 = A+u^\ell$. Then $[A,B] = [A,\ A+u^\ell)$. Therefore, proving that $\sigma \in [A,B]$ is equivalent to proving that $\sigma - A \in [0,u^\ell)$.

Now suppose that $u^{\ell-1} < B+1-A < u^\ell$ and that $0 < A < B$. The following inequalities relation is thus obtained:

$$B+1-u^\ell < A < B < B+1 < A+u^\ell.$$

Hence, $[A,B] = [A,B+1) = [A,\ A+u^\ell) \cap [B+1-u^\ell,\ B+1)$. Notice the absence of inequality between B and u^ℓ. This means that $u^\ell < B$ may occur and therefore it might not be possible to decompose σ as $\sigma = \sum_{j=0}^{\ell-1} (u^j \sigma_j)$. Furthermore, it is important to note that even if $A < \sigma < u^\ell$, this does not imply that for all $j \in \mathbb{Z}_u$, $\alpha_j < \sigma_j$, where α_j and σ_j are, respectively, the decomposition of A and σ.

To show that $\sigma \in [A,B]$, it suffices to show that $(\sigma - A)$ and $\left(\sigma + u^\ell - B - 1\right)$ are both in $[0,u^\ell)$.

Common Input: g, h, u, ℓ, A, B, and a commitment C.
Prover Input: σ, r such that $C = g^{\sigma} h^{r}$ and $\sigma \in [A, B]$.

$P \xleftarrow{y,\{A_i\}} V$
- Verifier picks $x \in_R \mathbb{Z}_p^*$ such that $-x \notin \mathbb{Z}_u$ and sends $y \leftarrow g^x$ and $A_i \leftarrow g^{\frac{1}{x+i}}$, $\forall i \in \mathbb{Z}_u$.
- Prover checks that $y \in \mathbb{G}_1^*$, $A_i \in \mathbb{G}_1^*$ and that $e(g,g) \stackrel{?}{=} e(A_i,\ y \cdot g^i)$, $\forall i \in \mathbb{Z}_u$.

$P \xrightarrow{\{V_j\},\{\widetilde{V_j}\}} V$
- Prover picks $v_j, \widetilde{v_j} \in_R \mathbb{Z}_p^*$, sends $V_j \leftarrow A_{\Upsilon_j}^{v_j}$ and $\widetilde{V_j} \leftarrow A_{\Psi_j}^{\widetilde{v_j}}$ for every $j \in \mathbb{Z}_\ell$, such that $(\sigma - A) = \sum_{j=0}^{\ell-1} \left(\Upsilon_j u^j\right)$, and $\left(\sigma + u^\ell - B - 1\right) = \sum_{j=0}^{\ell-1} \left(\Psi_j u^j\right)$.
- Verifier checks that $V_j, \widetilde{V_j} \in \mathbb{G}_1^*,\ \forall j \in \mathbb{Z}_\ell$.

Prover and Verifier run

$$\mathrm{PK}\left\{ \left(\{\Upsilon_j\},\{\Psi_j\},r,\{v_j\},\{\widetilde{v_j}\}\right): \begin{array}{lllll} Cg^{-A} = h^r g^{\sum_{j=0}^{\ell-1}(\Upsilon_j u^j)} & \wedge & Cg^{u^\ell - B - 1} = h^r g^{\sum_{j=0}^{\ell-1}(\Psi_j u^j)} \\ \wedge \quad V_j = g^{\frac{v_j}{x+\Upsilon_j}} & & \wedge \quad \widetilde{V_j} = g^{\frac{\widetilde{v_j}}{x+\Psi_j}}, & \forall j \in \mathbb{Z}_\ell \end{array} \right\}$$

$P \xrightarrow{\{a_j\},\{\widetilde{a_j}\},D} V$
- Prover picks $s_j, t_j, \widetilde{t_j}, m \in_R \mathbb{Z}_p^*$ for every $j \in \mathbb{Z}_\ell$ and sends $D \leftarrow h^m g^{\sum_{j=0}^{\ell-1}(s_j u^j)}$, $a_j \leftarrow e\left(V_j^{-s_j} g^{t_j}, g\right)$, and $\widetilde{a_j} \leftarrow e\left(\widetilde{V_j}^{-s_j} g^{\widetilde{t_j}}, g\right)$.
- Verifier checks that $a_j, \widetilde{a_j} \in \mathbb{G}_\mathrm{T},\ \forall j \in \mathbb{Z}_\ell$, and $D \in \mathbb{G}_1^*$.

$P \xleftarrow{c} V$
- Verifier sends a random challenge $c \in_R \mathbb{Z}_p^*$.
- Prover checks that $c \in \mathbb{Z}_p^*$.

$P \xrightarrow[\left\{z_{v_j}\right\},\left\{z_{\widetilde{v_j}}\right\}]{\left\{z_{\Upsilon_j}\right\},\left\{z_{\Psi_j}\right\},z_r} V$
- Prover sends $z_r \leftarrow (m - rc)$, $z_{\Upsilon_j} \leftarrow (s_j - \Upsilon_j c)$, $z_{\Psi_j} \leftarrow (s_j - \Psi_j c)$, $z_{v_j} \leftarrow (t_j - v_j c)$ and $z_{\widetilde{v_j}} \leftarrow (\widetilde{t_j} - \widetilde{v_j} c)$ for every $j \in \mathbb{Z}_\ell$.
- Verifier checks for every $j \in \mathbb{Z}_\ell$ that $z_{\Upsilon_j}, z_{\Psi_j}, z_{v_j}, z_{\widetilde{v_j}}, z_r \in \mathbb{Z}_p^*$, that $D \stackrel{?}{=} C^c g^{-Ac} h^{z_r} g^{\sum_{j=0}^{\ell-1}\left(z_{\Upsilon_j} u^j\right)}$, that $D \stackrel{?}{=} C^c g^{\left(u^\ell - B - 1\right)c} h^{z_r} g^{\sum_{j=0}^{\ell-1}\left(z_{\Psi_j} u^j\right)}$, that $a_j \stackrel{?}{=} e\left(V_j, y\right)^c \cdot e\left(V_j^{-z_{\Upsilon_j}} g^{z_{v_j}}, g\right)$ and that $\widetilde{a_j} \stackrel{?}{=} e\left(\widetilde{V_j}, y\right)^c \cdot e\left(\widetilde{V_j}^{-z_{\Psi_j}} g^{z_{\widetilde{v_j}}}, g\right)$.

Protocol 4.2 – Interactive range proof protocol for range $[A, B]$, with AND composition

Note that $(\sigma - A)$ can always be decomposed in base u with ℓ digits as $0 \leqslant (\sigma - A) < B + 1 - A < u^\ell$. Similarly, $\left(\sigma + u^\ell - B - 1\right)$ can also always be decomposed with ℓ digits as:

$$\begin{aligned} &0 < u^\ell - (B + 1 - A), \\ &0 < u^\ell + A - B - 1 \leqslant \left(\sigma + u^\ell - B - 1\right), \\ &0 < u^\ell - 1 + \sigma - B \leqslant u^\ell - 1 < u^\ell. \end{aligned}$$

Furthermore, the u signatures and the verification key need to be sent only once for both subsets. The resulting proof can be achieved with an AND composition as shown in Protocol 4.2.

The major modification compared to the use of two distinct basic range proofs, is the use of a same challenge c, with identical elements D and z_r. As both A and B are public, the same commitment C can be used for both subsets. Furthermore, the verification checks on D by the verifier need to be adapted as well. The first check of the verifier, $D \stackrel{?}{=} C^c g^{-Ac} h^{z_r} g^{\sum_{j=0}^{\ell-1}\left(z_{\Upsilon_j} u^j\right)}$, ensures that $(\sigma - A) = \sum_{j=0}^{\ell-1}\left(\Upsilon_j u^j\right)$. Similarly, the second check $D \stackrel{?}{=} C^c g^{(u^\ell - B - 1)c} h^{z_r} g^{\sum_{j=0}^{\ell-1}\left(z_{\Psi_j} u^j\right)}$ ensures that $\left(\sigma + u^\ell - B - 1\right) = \sum_{j=0}^{\ell-1}\left(\Psi_j u^j\right)$. The third check:

$$a_j \stackrel{?}{=} e\left(V_j, y\right)^c \cdot e\left(V_j^{-z_{\Upsilon_j}} g^{z_{v_j}}, g\right),$$

ensures that $0 \leqslant \Upsilon_j < u$. This implies that $0 \leqslant \sigma - A < u^\ell$ which is equivalent to showing that $\sigma \in [A,\ A + u^\ell)$. The fourth and last check:

$$\widetilde{a_j} \stackrel{?}{=} e\left(\widetilde{V_j}, y\right)^c \cdot e\left(\widetilde{V_j}^{-z_{\Psi_j}} g^{z_{\widetilde{v_j}}}, g\right),$$

ensures that $0 \leqslant \Psi_j < u$. This implies that $0 \leqslant \sigma + u^\ell - B - 1 < u^\ell$ which is equivalent to showing that $\sigma \in [B + 1 - u^\ell,\ B + 1)$. The latter concludes the range proof with an AND composition. With the additional sending of 4ℓ extra elements, the communication complexity becomes:

$$Com_{AND}(u, \ell) = 2\ell \cdot \left(\|\mathbb{G}_1\| + \|\mathbb{G}_\mathrm{T}\| + 2 \cdot \left\|\mathbb{Z}_p\right\|\right) + \left(\|\mathbb{G}_1\| + 2 \cdot \left\|\mathbb{Z}_p\right\|\right).$$

As for the computational cost of the prover, it will be dominated by $(6\ell + 2)$ exponentiations and 2ℓ pairings. The computational cost of the verifier will be dominated by $(4\ell + 7)$ exponentiations and 4ℓ pairings.

Corollary 4.2

If the u-Strong Diffie-Hellman assumption associated with a pairing generator $\mathsf{PG}\left(1^k\right)$ *holds, where k is the security parameter, then Protocol 4.2 is a zero-knowledge range argument for the range* $[A, B]$*, with communication complexity of* $O\left(\frac{k}{\log k - \log\log k}\right)$ *group elements.*

Proof

To show that Protocol 4.2 is a zero-knowledge range argument, three security properties need to be met: the *completeness* of the protocol, the *special soundness* property, and the *special honest verifier zero-knowledge* property.

1. Sim_{AND} retrieves $y, \{A_i\}$ from V^* (or from a trusted third party).

2. Sim_{AND} chooses $\sigma \in_R [A,B]$, $v_j, \widetilde{v_j} \in_R \mathbb{Z}_p^*$ for every $j \in \mathbb{Z}_\ell$,
   computes $V_j \leftarrow A_{\Upsilon_j}^{v_j}$ where $(\sigma - A) = \sum_{j=0}^{\ell-1} (\Upsilon_j u^j)$, and
   $\widetilde{V_j} \leftarrow A_{\Psi_j}^{\widetilde{v_j}}$ where $\left(\sigma + u^\ell - B - 1\right) = \sum_{j=0}^{\ell-1} (\Psi_j u^j)$.

3. Sim_{AND} runs the simulator of
$$\text{PK}\left\{ \left(\{\Upsilon_j\}, \{\Psi_j\}, r, \{v_j\}, \{\widetilde{v_j}\}\right) : \begin{array}{lll} Cg^{-A} = h^r g^{\sum_{j=0}^{\ell-1}(\Upsilon_j u^j)} & \wedge & Cg^{u^\ell - B - 1} = h^r g^{\sum_{j=0}^{\ell-1}(\Psi_j u^j)} \\ \wedge \quad V_j = g^{\frac{v_j}{x+\Upsilon_j}} & \wedge & \widetilde{V_j} = g^{\frac{\widetilde{v_j}}{x+\Psi_j}}, \quad \forall j \in \mathbb{Z}_\ell \end{array} \right\} :$$

   (a) On challenge $c \in \mathbb{Z}_p^*$, Sim_{AND} chooses $\{s_j\}, \{z_{v_j}\}, \{z_{\widetilde{v_j}}\}, z_r \in_R \mathbb{Z}_p^*$, and
   computes for every $j \in \mathbb{Z}_\ell$:
   $z_{\Upsilon_j} \leftarrow (s_j - \Upsilon_j c)$, and
   $z_{\Psi_j} \leftarrow (s_j - \Psi_j c)$.

   (b) Finally, Sim_{AND} computes $D \leftarrow C^c g^{-Ac} h^{z_r} g^{\sum_{j=0}^{\ell-1}\left(z_{\Upsilon_j} u^j\right)}$, and for every $j \in \mathbb{Z}_\ell$
   $a_j \leftarrow e(V_j, y)^c \cdot e(V_j, g)^{-z_{\Upsilon_j}} \cdot e(g,g)^{z_{v_j}}$, and
   $\widetilde{a_j} \leftarrow e(\widetilde{V_j}, y)^c \cdot e(\widetilde{V_j}, g)^{-z_{\Psi_j}} \cdot e(g,g)^{z_{\widetilde{v_j}}}$.

4. Sim_{AND} returns the transcript:
   $\left\{y, \{A_i\}, \{V_j\}, \{\widetilde{V_j}\}, \{a_j\}, \{\widetilde{a_j}\}, D, c, \{z_{\Upsilon_j}\}, \{z_{\Psi_j}\}, \{z_{v_j}\}, \{z_{\widetilde{v_j}}\}, z_r\right\}$.

Figure 4.2 – Simulator for the interactive range argument protocol, with AND composition

The *completeness* follows from the completeness property of Protocol 4.1 proven in Theorem 4.1 and from the decomposition in base u of $(\sigma - A)$ and $\left(\sigma + u^\ell - B - 1\right)$. As $0 \leqslant (\sigma - A) < u^\ell$ and $0 < \left(\sigma + u^\ell - B - 1\right) < u^\ell$, their decomposition consists of ℓ digits:

$$(\sigma - A) = \sum_{j=0}^{\ell-1} \left(\Upsilon_j u^j\right), \text{ such that } 0 \leqslant \Upsilon_j < u, \text{ and}$$
$$\left(\sigma + u^\ell - B - 1\right) = \sum_{j=0}^{\ell-1} \left(\Psi_j u^j\right), \text{ such that } 0 \leqslant \Psi_j < u.$$

Recall that for $B + 1 = u^\ell$, the AND decomposition is not needed as the range argument for $\sigma \in [A,B]$ is equivalent to the range argument for $(\sigma - A) \in [0, u^\ell)$.

The *special soundness* property follows from the special soundness of Protocol 4.1. The main difference in this case, is that the witness output by the extractor is $\left(\sigma, r, \{v_j\}, \{\widetilde{v_j}\}\right)$ and is computed as follows:

$$\sigma = A + \frac{\sum_{j=0}^{\ell-1}\left(z_{\Upsilon_j} - z'_{\Upsilon_j}\right) u^j}{c' - c}; \quad r = \frac{z_r - z'_r}{c' - c}; \quad v_j = \frac{z_{v_j} - z'_{v_j}}{c' - c}; \quad \widetilde{v_j} = \frac{z_{\widetilde{v_j}} - z'_{\widetilde{v_j}}}{c' - c}.$$

Identically to the case of Theorem 4.1, a malicious prover P^* that convinces a verifier can be (almost) directly used to perform a weak chosen-message attack against the Boneh-Boyen signature scheme. Valid signatures can be obtained with $V_j^{(1/v_j)}$ and with $\widetilde{V_j}^{(1/\widetilde{v_j})}$ for Υ_j and Ψ_j respectively. Recall that for every $j \in \mathbb{Z}_\ell$, $v_j \neq 0$ and $\widetilde{v_j} \neq 0$ for the same reasons as in the proof of Theorem 4.1 and of Theorem 3.2.

The *special honest verifier zero-knowledge* property is inherited from the same property of Protocol 4.1. The simulator Sim_{AND} of the interactions between any verifier V^* and any honest prover P is described in Figure 4.2. Although Sim_{AND} is slightly different from the simulator Sim (Figure 4.1), their conclusions are identical. The probability distribution of the transcript output by Sim_{AND} is identical to the one of a normal interaction between regular provers and verifiers. Note that the requirement $D \stackrel{?}{=} C^c g^{\left(u^\ell - B - 1\right)c} h^{z_r} g^{\sum_{j=0}^{\ell-1}\left(z_{\Psi_j} u^j\right)}$ is also met.■

Further optimization can be achieved if the additional restriction $B + 1 - A < 2u^{\ell-1}$ is met. Indeed, in this case the AND composition based on the range $[0, u^\ell)$ can be replaced with an OR composition (also known as OR-proof) based on the range $[0, u^{\ell-1})$. The decomposition becomes:

$$[A, B] = [A, B+1) = [A, A + u^{\ell-1}) \cup [B + 1 - u^{\ell-1}, B + 1).$$

The OR composition for proving that $\sigma \in [A, B]$ is directly obtained by applying the "proof of OR" presented by Cramer et al. in [CDS94] (Corollary 12 and 13) with Protocol 4.1 for the range $[0, u^{\ell-1})$. The drawback of this proof technique is that it imposes the weaker security property of witness hiding (see Section 2.3.5), instead of the zero-knowledge security property.

Corollary 4.3 ([CDS94](Corollary 12))
Let protocol Θ be a three round public coin, honest verifier zero-knowledge proof of knowledge for relation R, which satisfies the special soundness property. Then for any n, d there is a protocol with the same round complexity as Θ in which the prover shows that he knows d out of n witnesses without revealing which d witnesses are known.

Note that *public coin* means that the verifier sends solely a uniformly random challenge in the Σ-protocol.

Corollary 4.4 ([CDS94](Corollary 13))
Consider the protocol guaranteed by Corollary 4.3, let $n = 2$ and $d = 1$, i.e. the prover proves that he knows at least 1-out-of-2 solutions. For any generator G generating pairs in R, this protocol is witness hiding over G^2.

The resulting protocol consists of running two range proofs in parallel: an honest and a simulated one. The following explanations are provided for the case that $\sigma \in [A, A + u^{\ell-1})$. The alternative case is easily obtained by swapping the subsets for the honest and simulated sub-range proofs. The subset $[A, A + u^{\ell-1})$ will be called the *true subset*, as it contains σ.

1. Sim_s honestly retrieves $y, \{A_i\}$ from V (or from a trusted third party).
2. Sim_s chooses $\sigma \in_R [B+1-u^{\ell-1}, B+1)$, $v_j \in_R \mathbb{Z}_p^*$ for every $j \in \mathbb{Z}_{\ell-1}$, and computes $\widetilde{V_j} \leftarrow A_{\Psi_j}^{v_j}$ where $\left(\sigma + u^{\ell-1} - B - 1\right) = \sum_{j=0}^{\ell-2} \left(\Psi_j u^j\right)$.
3. Sim_s runs the simulator of
$$\text{PK}\left\{\left(\{\Psi_j\}, r, \{v_j\}\right) : Cg^{u^{\ell-1}-B-1} = h^r g^{\sum_{j=0}^{\ell-2}(\Psi_j u^j)} \ \wedge \ \widetilde{V_j} = g^{\frac{v_j}{x+\Psi_j}}, \ \forall j \in \mathbb{Z}_{\ell-1}\right\}:$$
   (a) Sim_s chooses $c_s, \{z_{\Psi_j}\}, \{\widetilde{z_{v_j}}\}, \widetilde{z_r} \in_R \mathbb{Z}_p^*$.
   (b) Finally, Sim_s computes $\widetilde{D} \leftarrow C^{c_s} g^{\left(u^{\ell-1}-B-1\right)c_s} h^{\widetilde{z_r}} g^{\sum_{j=0}^{\ell-2}\left(z_{\Psi_j} u^j\right)}$ and $\widetilde{a_j} \leftarrow e(\widetilde{V_j}, g)^{-z_{\Psi_j} - \Psi_j c_s} e(g,g)^{\widetilde{z_{v_j}} + v_j c_s}$ for every $j \in \mathbb{Z}_{\ell-1}$.
4. Sim_s returns the transcript $\left\{y, \{A_i\}, \{\widetilde{V_j}\}, \{\widetilde{a_j}\}, \widetilde{D}, c_s, \{z_{\Psi_j}\}, \{\widetilde{z_{v_j}}\}, \widetilde{z_r}\right\}$.

Figure 4.3 – Simulator for the simulated range argument protocol $\sigma \in [B+1-u^{\ell-1}, B+1)$

The subset $[B+1-u^{\ell-1}, B+1)$ does not necessarily contain σ and hence will be called the *simulated subset*. The simulated range proof corresponds to the *simulated subset*. In order to perform the simulated range proof, the prover will use the simulator Sim_s described in Figure 4.3. Note that y and $\{A_i\}$ are honestly retrieved from the verifier, but the challenge c_s is chosen by the prover. The elements $\{z_{\Psi_j}\}, \{\widetilde{z_{v_j}}\}, \widetilde{z_r}$ from the simulator Sim_s will be sent to the verifier only after the prover has retrieved the challenge c from the verifier. Additionally, the prover also sends $c_B \leftarrow c_s$. The honest range proof for the *true subset* is performed as described in Protocol 4.1 with minor modifications. The target range is $[0, u^{\ell-1})$ instead of $[0, u^{\ell})$, and in order to prove that $\sigma \in [A, A+u^{\ell-1})$, the digits σ_j are replaced with the digits Υ_j such that $(\sigma - A) = \sum_{j=0}^{\ell-2} \left(\Upsilon_j u^j\right)$. Hence the check of the verifier on D is modified as $D \stackrel{?}{=} C^{c_A} g^{-Ac_A} h^{z_r} g^{\sum_{j=0}^{\ell-2}\left(z_{\Upsilon_j} u^j\right)}$, where $c_A \leftarrow (c - c_s)$. The prover is also required to send c_A to the verifier, who will additionally check that $c \stackrel{?}{=} c_A + c_B$. To complete the OR composition, the verifier also needs to check that:

$$\widetilde{D} \stackrel{?}{=} C^{c_B} g^{\left(u^{\ell-1}-B-1\right)c_B} h^{\widetilde{z_r}} g^{\sum_{j=0}^{\ell-2}\left(z_{\Psi_j} u^j\right)}, \text{ and}$$
$$\widetilde{a_j} \stackrel{?}{=} e\left(\widetilde{V_j}, y\right)^{c_B} \cdot e\left(\widetilde{V_j}^{-z_{\Psi_j}} g^{\widetilde{z_{v_j}}}, g\right).$$

The range proof with OR composition requires less elements to be sent compared to the AND composition, even thought the two challenges c_A and c_B need to be sent:

$$\begin{aligned} Com_{OR}(u,\ell) &= 2 \cdot Com(u, \ell-1) - \left\|\mathbb{Z}_p\right\| + 2 \cdot \left\|\mathbb{Z}_p\right\| \\ &= 2 \cdot \left[(\ell-1) \cdot \left(\|\mathbb{G}_1\| + \|\mathbb{G}_\mathrm{T}\| + 2 \cdot \left\|\mathbb{Z}_p\right\|\right) + \left(\|\mathbb{G}_1\| + 2 \cdot \left\|\mathbb{Z}_p\right\|\right)\right] + \left\|\mathbb{Z}_p\right\| \\ &= 2\ell \cdot \left(\|\mathbb{G}_1\| + \|\mathbb{G}_\mathrm{T}\| + 2 \cdot \left\|\mathbb{Z}_p\right\|\right) + \left(\left\|\mathbb{Z}_p\right\| - 2 \cdot \|\mathbb{G}_\mathrm{T}\|\right). \end{aligned}$$

The gain for the communication load is of $\left(\|\mathbb{G}_1\| + 2 \cdot \|\mathbb{G}_T\| + \left\|\mathbb{Z}_p\right\|\right)$ bits, when compared to the AND composition. The prover computational cost will be dominated by 6ℓ exponentiations and $2(\ell - 1)$ pairings. Finally, the verifier computational cost will be dominated by $(4\ell + 6)$ exponentiations and $4(\ell - 1)$ pairings.

Corollary 4.5

If the u-Strong Diffie-Hellman assumption associated with a pairing generator $\mathsf{PG}\left(1^k\right)$ *holds, where k is the security parameter, then the OR composition described above is a witness hiding range argument for the range* $[A, B]$*, with communication complexity of* $O\left(\frac{k}{\log k - \log\log k}\right)$ *group elements.*

Proof

Theorem 4.1, Corollary 4.3, and Corollary 4.4 imply that the OR composition is a witness hiding range argument. Let Θ be the Protocol 4.1. Then Θ is a public coin protocol as the verifier sends only a random challenge. Furthermore, it satisfies honest verifier zero-knowledge and special soundness (Theorem 4.1). Hence Θ meets the requirements from Corollary 4.3. Therefore, from Corollary 4.4, the OR composition is a witness hiding range argument. ■

Last but not least, proving that $\sigma \in [A, B]$ where $A < 0$ can be reduced to one of the previous cases by showing that $\sigma - A \in [0, B - A]$.

Common Input:	u, ℓ, and a commitment C;
	the commitment scheme parameters $\mathsf{Param}_{\mathsf{com}}$;
	and the set membership proof parameters $\mathsf{Param}_{\mathsf{smem}}$.
Prover Input:	σ, r such that $C = \mathsf{Commit}(\sigma, r)$ and $\sigma \in [0, u^\ell)$.

Prover and Verifier run

$$\mathrm{PK}\left\{\left(\{\sigma_j\}, r\right) : C = h^r g^{\sum_{j=0}^{\ell-1}(\sigma_j u^j)} \ \wedge \ \sigma_j \in \mathbb{Z}_u, \ \forall j \in \mathbb{Z}_\ell\right\}$$

Protocol 4.3 – Interactive range proof protocol for range $[0, u^\ell)$, from a general set membership proof

Using Alternative Set Membership Proof. Protocol 4.1 can be adapted to use any appropriate alternative set membership proof. Moreover, when combined with a generic set membership proof, a generalization of the u-ary decomposition for range proofs is obtained. This is described in Protocol 4.3. The prover needs to show that his secret element can be decomposed in ℓ digits, and that each of these digits is in $\mathbb{Z}_u$ using the generic set membership proof. The range proof $\sigma \in [0, u^\ell)$ is thus obtained from the following proof of knowledge:

$$\mathrm{PK}\left\{\left(\{\sigma_j\}, r\right) : C = h^r g^{\sum_{j=0}^{\ell-1}(\sigma_j u^j)} \ \bigwedge \ \sigma_j \in \mathbb{Z}_u, \ \forall j \in \mathbb{Z}_\ell\right\}.$$

In order to achieve overall minimal communication complexity, the communication load of the set membership proof will dictate the values of parameters u and ℓ.

An alternative set membership proof example is to use a variant of the set membership proof of Arfaoui et al. from [ALT+15a], which is essentially a Boneh-Boyen based set membership proof without pairings. This variant, although exclusively introduced in this thesis, can be easily deduced from [ALT+15a]. The resulting range proof is described in Protocol 4.4, where the boxed elements correspond to the elements that differ from Protocol 4.1. Regarding the underlying set membership proof, the main difference with the original version of Arfaoui et al. is the way signatures are verified. Protocol 4.4 uses pairings for checking the validity of the signatures, thus bilinear groups and associated computational hardness assumptions are still needed. This requirement can be relaxed, but will induce a high efficiency cost. Indeed, in the original protocol of Arfaoui et al., the verification of the signatures A_i is performed with the proof of knowledge described in Figure 4.4, repeated for every signature A_i. The

1. The prover (V in the range proof) randomly picks $s_i \in_R \mathbb{Z}_p$.
2. Prover V sends $a_i \leftarrow A_i^{s_i}$ and $b_i \leftarrow g^{s_i}$ to the verifier P.
3. Verifier P sends a challenge $c_i \in_R \mathbb{Z}_p$.
4. Prover V sends the reply $r_i \leftarrow s_i + c_i \cdot x$ to the verifier P.
5. Verifier P checks that $A_i^{r_i} \stackrel{?}{=} a_i \left(g A_i^{-i}\right)^{c_i}$ and that $g^{r_i} \stackrel{?}{=} b_i y^{c_i}$.

Figure 4.4 – Proof of knowledge $\mathrm{PK}\{(x) : y = g^x \ \wedge \ A_i^x = g \cdot A_i^{-i}\}$

variant of their method improves both the computational complexity and the communication load. Both the prover and the verifier no longer need to compute bilinear pairings in the range proof. Hence the overall computational complexity of the prover is reduced to $(3\ell + 2)$ exponentiations, and that of the verifier is reduced to $(2\ell + 3)$ exponentiations. Regarding the communication complexity, the elements $a_j \in \mathbb{G}_\mathrm{T}$ are all replaced with the elements $E_j \in \mathbb{G}_1$. Thus the communication load becomes:

$$Com(u, \ell) = 2\ell \cdot \left(\|\mathbb{G}_1\| + \left\|\mathbb{Z}_p\right\|\right) + \left(\|\mathbb{G}_1\| + 2 \cdot \left\|\mathbb{Z}_p\right\|\right).$$

Note, however, that this communication complexity holds for the basic range proof $\sigma \in [0, u^\ell)$. In order to handle arbitrary ranges, the same technique described before should be used, namely the AND composition (see Protocol 4.2). Thus the communication load for arbitrary ranges $[A, B]$ becomes:

$$Com_{AND}(u, \ell) = 4\ell \cdot \left(\|\mathbb{G}_1\| + \left\|\mathbb{Z}_p\right\|\right) + \left(\|\mathbb{G}_1\| + 2 \cdot \left\|\mathbb{Z}_p\right\|\right).$$

Regarding the computational cost of the AND composition, the computational cost of the prover will be dominated by $(6\ell + 2)$ exponentiations, and that of the verifier will be dominated by $(4\ell + 6)$ exponentiations.

Common Input: g, h, u, ℓ, and a commitment C.
Prover Input: σ, r such that $C = g^{\sigma} h^{r}$ and $\sigma \in [0, u^{\ell})$.

$P \xleftarrow{y,\{A_i\}} V$
- Verifier picks $x \in_R \mathbb{Z}_p^*$ such that $-x \notin \mathbb{Z}_u$ and sends $y \leftarrow g^x$ and $A_i \leftarrow g^{\frac{1}{x+i}}, \forall i \in \mathbb{Z}_u$.
- Prover checks that $y \in \mathbb{G}_1^*$, $A_i \in \mathbb{G}_1^*$ and that $e(g,g) \stackrel{?}{=} e(A_i,\ y \cdot g^i), \forall i \in \mathbb{Z}_u$.

$P \xrightarrow{\{V_j\}} V$
- Prover picks $v_j \in_R \mathbb{Z}_p^*$ and sends $V_j \leftarrow A_{\sigma_j}^{v_j}$, for every $j \in \mathbb{Z}_\ell$, such that $\sigma = \sum_{j=0}^{\ell-1} (\sigma_j u^j)$.
- Verifier checks that $V_j \in \mathbb{G}_1^*,\ \forall j \in \mathbb{Z}_\ell$.

Prover and Verifier run

$$\mathrm{PK}\left\{ \left(\{\sigma_j\}, r, \{v_j\}\right) :\ C = h^r g^{\sum_{j=0}^{\ell-1}(\sigma_j u^j)} \ \wedge\ V_j = g^{\frac{v_j}{x+\sigma_j}},\ \forall j \in \mathbb{Z}_\ell \right\}$$

$P \xrightarrow{\boxed{\{E_j\}},D} V$
- Prover picks $s_j, t_j, m \in_R \mathbb{Z}_p^*$ for every $j \in \mathbb{Z}_\ell$ and sends $\boxed{E_j \leftarrow V_j^{-s_j} g^{t_j}}$ and $D \leftarrow h^m g^{\sum_{j=0}^{\ell-1}(s_j u^j)}$.
- Verifier checks that $\boxed{E_j}, D \in \mathbb{G}_1^*,\ \forall j \in \mathbb{Z}_\ell$.

$P \xleftarrow{c} V$
- Verifier sends a random challenge $c \in_R \mathbb{Z}_p^*$.
- Prover checks that $c \in \mathbb{Z}_p^*$.

$P \xrightarrow{\{z_{\sigma_j}\},\{z_{v_j}\},z_r} V$
- Prover sends $z_r \leftarrow (m - rc)$, and $z_{\sigma_j} \leftarrow (s_j - \sigma_j c),\ z_{v_j} \leftarrow (t_j - v_j c)$ for every $j \in \mathbb{Z}_\ell$.
- Verifier checks that $z_{\sigma_j}, z_{v_j}, z_r \in \mathbb{Z}_p^*$ for every $j \in \mathbb{Z}_\ell$, that $D \stackrel{?}{=} C^c h^{z_r} g^{\sum_{j=0}^{\ell-1}\left(z_{\sigma_j} u^j\right)}$ and that $\boxed{E_j \stackrel{?}{=} V_j^{cx - z_{\sigma_j}} g^{z_{v_j}}}$ for every $j \in \mathbb{Z}_\ell$.

Protocol 4.4 – Interactive range proof protocol, based on Arfaoui et al. set membership proof

Theorem 4.6

If the u-Strong Diffie-Hellman assumption associated with a pairing generator PG *holds, then Protocol 4.4 is a zero-knowledge range argument for the range* $[0, u^\ell)$.

Proof

The proof to show that Protocol 4.4 is a zero-knowledge range argument, is identical to the proof of Theorem 4.1 with minor modifications. *Completeness* is achieved, as the following holds:

$$\begin{aligned} V_j^{cx-z_{\sigma_j}} g^{z_{v_j}} &= V_j^{cx-s_j+c\sigma_j} g^{t_j-cv_j} \\ &= V_j^{c(x+\sigma_j)} V_j^{-s_j} g^{-cv_j} g^{t_j} \\ &= g^{\frac{v_j}{(x+\sigma_j)} \cdot c(x+\sigma_j)} g^{-cv_j} V_j^{-s_j} g^{t_j} \\ &= V_j^{-s_j} g^{t_j} \\ &= E_j. \end{aligned}$$

Special soundness is achieved in the same way as in Theorem 4.1. The simulator for the *special honest verifier zero-knowledge* is achieved as described in Figure 4.1, with a modification in step 3b: the computations of a_j are replaced with the computations of $E_j \leftarrow V_j^{-z_{\sigma_j}-\sigma_j c} g^{z_{v_j}+v_j c}$. ■

4.4 Sumset Representation of Integer Intervals

Using the u-ary decomposition, handling arbitrary ranges $[A, B]$ almost doubles the communication load as well as the computational complexity when compared to the basic $[0, u^\ell)$ range proof. This is essentially due to the use of two basic range proofs in order achieve the flexibility of arbitrary ranges. A better solution is therefore achieved when the range decomposition allows some flexibility in the base decomposition, as is the case with multi-base decomposition. Such decomposition can be obtained with a sumset representation of integer intervals. Informally, although this decomposition uses u-ary digits, the base corresponding to the digits is more flexible. For $\sigma \in [0, H]$, the following explains how to derive the sumset representation $\sigma = \omega' + \sum_{i=0}^{\ell-1} \sigma_i G_i$, where G_i are public parameters, $\sigma_i \in \mathbb{Z}_u$, $\omega' \in [0, H']$, $0 \leqslant H' < u-1$, and $2 \leqslant u \ll H$. Here, $H' = \left(H - (u-1) \cdot \left\lfloor \frac{H}{u-1} \right\rfloor\right)$ as all sub-intervals $[0, (u-1)]$ are already included by the elements σ_i. Moreover, the aim is to find minimal ℓ for any fixed value of H and u. Handling arbitrary ranges $[A, B]$ will be achieved by proving that $\sigma - A \in [0, H]$, where $H = B - A$.

An intuitive explanation of the sumset representation will be provided first. Suppose that H and u are fixed, and let $H_0 = H$. Then clearly $\sigma \in [0, H_0]$ and $\sigma = \omega_1 + \sigma_0 G_0$, where $\sigma_0 \in \mathbb{Z}_u$, $\omega_1 \in [0, H_1] = [0, H_0 - (u-1)G_0]$, and G_0 is defined as $G_0 := \lfloor (1 + H_0)/u \rfloor$. This can be derived from the goal to divide $[0, H_0]$ into u smaller (possibly overlapping) intervals of equal size H_1 such that H_1 is minimal. The sub-intervals should start at periodic positions jG_0, for

some G_0 and for $0 \leqslant j \leqslant (u-1)$. Ideally, the start of each new sub-intervals should be just after the end of each previous sub-intervals. This would imply that $G_0 = 1 + H_1$. To guarantee overlapping and ensure that there are no isolated elements between sub-intervals, the start of each sub-intervals should be before or equal to the ideal case. Therefore, $G_0 \leqslant 1 + H_1$. Furthermore, in order to reach the upper limit H_0, the following condition must also holds: $H_0 = (u-1)G_0 + H_1$. Thus, the optimal case when $H_1 = G_0 - 1$, implies that $H_0 = uG_0 - 1$ or $G_0 = (1 + H_0)/u$. Since G_0 has to be an integer, it has to be set as $G_0 = \lfloor (1 + H_0)/u \rfloor$. Finally, as stated above, $H_1 = H_0 - (u-1)G_0$.

These formulas reduce the case $[0, H_0]$ to a smaller case $[0, H_1]$ that can be solved similarly. Recursively, $\sigma = \omega' + \sum_j \sigma_j G_j$, with $\omega' \in [0, H']$, $\sigma_j \in \mathbb{Z}_u$, and where

$$G_j = \left\lfloor \frac{1 + H_j}{u} \right\rfloor, \text{ and} \tag{4.6}$$

$$\begin{aligned} H_{j+1} &= H_j - (u-1)G_j \\ &= H_j - \left\lfloor \frac{1 + H_j}{u} \right\rfloor \cdot (u-1). \end{aligned} \tag{4.7}$$

This process is carried out for as long as $H_j \geqslant u-1$. It stops when the interval $[0, H_{j+1}]$ is small enough so that it cannot be covered by u different non-empty intervals, that is, if $H_{j+1} < u-1$. At that point and as the recursive process is completed, the number ℓ of steps in this recursive process can be defined as $\ell(u, H) := j + 1$. Furthermore, H' can be set as $H' = H_\ell = H_{j+1}$.

For example, with $H = 57$ (thus $\sigma \in [0,57]$) and $u = 4$ ($\sigma_i \in [0,3]$), it can be verified that $\sigma = 14\sigma_0 + 4\sigma_1 + \sigma_2$. As $(4-1) \mid 57$, this implies that $H' = 0$. By way of providing a further example with $H = 160$ and $u = 4$:

$$\begin{aligned} \sigma &= 40\sigma_0 + \omega_1, \text{ with } \omega_1 \in [0,40] \\ &= 40\sigma_0 + 10\sigma_1 + \omega_2, \text{ with } \omega_2 \in [0,10] \\ &= 40\sigma_0 + 10\sigma_1 + 2\sigma_2 + \omega_3, \text{ with } \omega_3 \in [0,4] \\ &= 40\sigma_0 + 10\sigma_1 + 2\sigma_2 + \sigma_3 + \omega', \text{ with } \omega' \in [0,1]. \end{aligned}$$

The recursive process is now complete since $1 < u - 1 = 3$.

Lemma 4.7

Let H_j and G_j be defined respectively as in equations (4.7) *and* (4.6)*, then the sequence H_j is a finite monotone (strictly) decreasing sequence, the sequence G_j is a finite monotonic decreasing sequence ($G_{j+1} \leqslant G_j$), and $0 \leqslant H_\ell \leqslant u-2$, where H_ℓ is the last element of the sequence H_j.*

Proof

Showing that the sequence H_j is monotone decreasing, can be proved by induction. In order to do so, two steps need to be demonstrated: $H_1 < H_0$ and $H_{j+1} < H_j$.

First, consider the case of H_1:

$$\begin{aligned}
H_1 &= H_0 - (u-1)G_0 \\
&= H_0 - (u-1)\cdot\left\lfloor\frac{1+H_0}{u}\right\rfloor \\
&< H_0 - (u-1)\left(\frac{1+H_0}{u} - 1\right) \\
&< H_0 - (u-1)\left(\frac{1+H_0-u}{u}\right) \\
&< H_0 - \left(\frac{u-1}{u}\right)(H_0-(u-1)) \\
&< H_0.
\end{aligned}$$

The last inequality is achieved as $H_0 > (u-1)$ and $u \geqslant 2$. Thus, for all $j < \ell$ and as $H_j \geqslant (u-1)$, the following holds:

$$\begin{aligned}
H_{j+1} &= H_j - (u-1)G_j \\
&= H_j - (u-1)\cdot\left\lfloor\frac{1+H_j}{u}\right\rfloor \\
&< H_j - (u-1)\left(\frac{1+H_j}{u} - 1\right) \\
&< H_j - (u-1)\left(\frac{1+H_j-u}{u}\right) \\
&< H_j - \left(\frac{u-1}{u}\right)\big(H_j-(u-1)\big) \\
&< H_j.
\end{aligned}$$

As $H_{j+1} < H_j$, then the sequence H_j is monotone decreasing. This sequence is also finite as it stops when $H_\ell < u-1$.

Thus, regarding the sequence G_j, the following holds:

$$G_{j+1} = \left\lfloor\frac{1+H_{j+1}}{u}\right\rfloor \leqslant \left\lfloor\frac{1+H_j}{u}\right\rfloor = G_j.$$

Thus, the sequence G_j is a monotonic decreasing sequence. It is also finite as it has one element less than the sequence H_j.

Recall that the sequence H_j stops when $H_\ell < u-1$, therefore $H_\ell \leqslant u-2$. Furthermore, H_ℓ is necessarily positive:

$$\begin{aligned} H' = H_\ell &= H_{\ell-1} - (u-1)\cdot\left\lfloor \frac{1+H_{\ell-1}}{u} \right\rfloor \\ &\geqslant H_{\ell-1} - (u-1)\left(\frac{1+H_{\ell-1}}{u}\right) \\ &\geqslant (uH_{\ell-1} - (u-1)(1+H_{\ell-1}))\cdot u^{-1} \\ &\geqslant (uH_{\ell-1} - (u-1) - (u-1)H_{\ell-1})\cdot u^{-1} \\ &\geqslant (H_{\ell-1} - (u-1))\cdot u^{-1} \\ &\geqslant 0. \end{aligned}$$

This concludes the proof. ■

The following lemma gives an upper bound for the number of steps in the decomposition process. This result is slightly better than the one presented in [CLs10], as shown in Corollary 4.9.

Lemma 4.8

In the case of a range $[0, H]$*, and its decomposition with* H_j *and* G_j *are defined respectively as in equations* (4.7) *and* (4.6)*, then the decomposition process is guaranteed to stop in* ℓ *steps, where* $\ell = \ell(u,H) \leqslant 1 + \log_u(H-(u-2)) - \log_u 2$.

Proof

Recall that the decomposition process stops when $H_\ell < u-1$. This implies that $H_{\ell-1} \geqslant u-1 \geqslant 2u^0 + (u-2)$. Furthermore, it can be shown by induction that $H_j \geqslant 2u^{\ell-j-1} + (u-2)$. In that case, the first step consists of supposing that $H_{j+1} \geqslant 2u^{\ell-j-2} + (u-2)$. Therefore, the following implications unfold:

$$\begin{aligned} H_j &= H_{j+1} + (u-1)G_j \\ &= H_{j+1} + (u-1)\cdot\left\lfloor \frac{1+H_j}{u} \right\rfloor \\ &\geqslant 2u^{\ell-j-2} + (u-2) + (u-1)\left(\frac{1+H_j-(u-1)}{u}\right) \\ &\Longrightarrow uH_j \geqslant 2u^{\ell-j-1} + u(u-2) + (u-1)H_j - (u-1)(u-2) \\ &\qquad\geqslant 2u^{\ell-j-1} + (u-2) + (u-1)H_j \\ &\Longrightarrow H_j \geqslant 2u^{\ell-j-1} + (u-2). \end{aligned}$$

As $H = H_0$, it implies that $H \geqslant 2u^{\ell-1} + (u-2)$. Therefore:

$$\begin{aligned} & H \geqslant 2u^{\ell-1} + (u-2) \\ & \implies u^{\ell-1} \leqslant \frac{H-(u-2)}{2} \\ & \implies u^{\ell} \leqslant \frac{u}{2}(H-(u-2)) \\ & \implies \ell \leqslant \log_u\left(\frac{u}{2}(H-(u-2))\right) \\ & \implies \ell \leqslant 1 + \log_u(H-(u-2)) - \log_u 2. \end{aligned}$$

■

Corollary 4.9

In the case of a range $[0,H]$*, its decomposition process is guaranteed to stop in* $\ell \leqslant \lceil \log_u(H) \rceil$ *steps. Moreover, the case* $H \leqslant u^L$ *implies that* $\ell \leqslant L$*.*

Proof

The proof follows from Lemma 4.8. First, note that the following relations hold:

$$\begin{aligned} \ell \leqslant 1 + \log_u(H-(u-2)) - \log_u 2 &= 1 + \log_u(H+2-u) - \log_u 2 \\ &< 1 + \log_u(H+2-u) \leqslant 1 + \log_u(H). \end{aligned}$$

Thus $\ell < 1 + \log_u(H)$ implies that $\ell \leqslant \lceil \log_u(H) \rceil$. Therefore $H \leqslant u^L$ implies that $\ell \leqslant L$. ■

Lemma 4.7 and 4.8 lead to the following theorem:

Theorem 4.10

Let $u \geqslant 2$*,* $H \geqslant u$*. Let* G_j, H_j *be defined respectively as in equations* (4.6) *and* (4.7)*. Let* H' *be defined as before (*$H' < u-1$*). Denote* $\ell = \ell(u,H)$ *as defined previously. The sumset representation of* $\sigma \in [0,H]$ *is therefore* $\sigma = \omega' + \sum_{j=0}^{\ell-1} \sigma_j G_j$*, where* $\sigma_j \in \mathbb{Z}_u$*,* $\omega' \in [0,H']$*, and* $\ell \leqslant 1 + \log_u(H-(u-2)) - \log_u 2$*. Furthermore, if* $(u-1) \mid H$ *then* $H' = 0$*.*

Proof

Recall that $H' = H_\ell$. The decomposition is provided with equations (4.6) and (4.7). Lemma 4.7 proves that the decomposition is complete. Lemma 4.8 provides the proof for the upper bound on ℓ. As $H' = H - (u-1) \cdot \left\lfloor \frac{H}{u-1} \right\rfloor$, it implies that if $(u-1) \mid H$ then $H' = 0$. ■

Semi-Closed Form for G_j and H_j. While the presented recursive formulas for G_j and H_{j+1} are efficient, it is desirable to have a closed form for both of them. The following construction achieves semi-closed forms, which are formulas that only depend on u, j, and H.

Assume the basic u-ary decomposition $H = \sum_{j=0}^{L-1} h_j u^j$ with $h_j \in \mathbb{Z}_u$. For any $j \leqslant L$, write $\widehat{h_j} := \lfloor H/u^j \rfloor$, that is, $H = u^j \widehat{h_j} + \sum_{i=0}^{j-1} u^i h_i$. This implies the following:

$$
\begin{aligned}
H &= u^j \widehat{h_j} + \sum_{i=0}^{j-1} u^i h_i && (4.8)\\
&\Longrightarrow u^j \widehat{h_j} + \sum_{i=0}^{j-1} u^i h_i = u^{j+1} \widehat{h_{j+1}} + \sum_{i=0}^{j} u^i h_i \\
&\Longrightarrow u^j \widehat{h_j} = u^{j+1} \widehat{h_{j+1}} + u^j h_j \\
&\Longrightarrow \widehat{h_j} = u \widehat{h_{j+1}} + h_j. && (4.9)
\end{aligned}
$$

Define the notation $[\![x]\!] := r$, such that $0 \leqslant r < u-1$ and $r \equiv x \pmod{(u-1)}$. It is important for the theorem bellow, to keep in mind the following properties resulting from this notation:

$$
\begin{aligned}
(u-1) \mid a &\Longleftrightarrow [\![a]\!] = 0 \\
(u-1) \mid a &\Longleftrightarrow [\![a+b]\!] = [\![b]\!] \\
[\![a]\!] = 0 &\Longleftrightarrow [\![a+b]\!] = [\![b]\!] \\
[\![a]\!] = a &\Longleftrightarrow 0 \leqslant a < (u-1) \\
[\![a+b]\!] &= [\![a+[\![b]\!]]\!] \\
0 \leqslant a + [\![b]\!] < (u-1) &\Longleftrightarrow [\![a+b]\!] = a + [\![b]\!].
\end{aligned}
$$

Theorem 4.11

Let G_j, H_j be defined respectively as in equations (4.6) *and* (4.7)*, let $\widehat{h_j} = \lfloor H/u^j \rfloor$ and $H = \sum_{j=0}^{L-1} h_j u^j$, then:*
$H_j = \widehat{h_j} + \left[\!\!\left[\sum_{i=0}^{j-1} h_i \right]\!\!\right]$ *for* $0 < j \leqslant \ell$, *and*
$G_j = \widehat{h_{j+1}} + \left\lfloor \frac{1 + h_j + [\![\sum_{i=0}^{j-1} h_i]\!]}{u} \right\rfloor$, *for* $0 < j < \ell$.

Proof

The proof is achieved by induction. This is accomplished by showing the veracity of the induction basis ($j = 1$) and then proving the induction step for $j \geqslant 1$.

First, $H_0 = \widehat{h_0}$ can be derived from the definition of $\widehat{h_j} = \left\lfloor \frac{H}{u^j} \right\rfloor$. As $\widehat{h_j} = u\widehat{h_{j+1}} + h_j$, G_0 can be reformulated as

$$
\begin{aligned}
G_0 &= \left\lfloor \frac{1 + H_0}{u} \right\rfloor = \left\lfloor \frac{1 + \widehat{h_0}}{u} \right\rfloor \\
&= \left\lfloor \frac{1 + u\widehat{h_1} + h_0}{u} \right\rfloor \\
&= \widehat{h_1} + \left\lfloor \frac{1 + h_0}{u} \right\rfloor.
\end{aligned}
$$

Thus, H_1 can be derived as follows:

$$\begin{aligned} H_1 &= H_0 - (u-1)G_0 = \widehat{h_0} - (u-1)G_0 = \\ &= \left(u\widehat{h_1} + h_0\right) - (u-1)\left(\widehat{h_1} + \left\lfloor \frac{1+h_0}{u} \right\rfloor\right) \\ &= \widehat{h_1} + h_0 - (u-1)\left\lfloor \frac{1+h_0}{u} \right\rfloor \\ &= \widehat{h_1} + [\![h_0]\!] \end{aligned}$$

For the last equality, if $h_0 < (u-1)$ then $h_0 - (u-1)\left\lfloor \frac{1+h_0}{u} \right\rfloor = h_0 = [\![h_0]\!]$ and if $h_0 = (u-1)$ then $h_0 - (u-1)\left\lfloor \frac{1+h_0}{u} \right\rfloor = (u-1) - (u-1) = 0 = [\![h_0]\!]$. Now, G_1 can be derived as follows:

$$\begin{aligned} G_1 &= \left\lfloor \frac{1+H_1}{u} \right\rfloor \\ &= \left\lfloor \frac{1+\widehat{h_1}+[\![h_0]\!]}{u} \right\rfloor \\ &= \left\lfloor \frac{1+u\widehat{h_2}+h_1+[\![h_0]\!]}{u} \right\rfloor \\ &= \widehat{h_2} + \left\lfloor \frac{1+h_1+[\![h_0]\!]}{u} \right\rfloor. \end{aligned}$$

This concludes the induction basis. The induction step hypothesis assumes that

$$\begin{aligned} H_j &= \widehat{h_j} + \left[\!\left[\sum_{i=0}^{j-1} h_i \right]\!\right] \\ &= u\widehat{h_{j+1}} + h_j + \left[\!\left[\sum_{i=0}^{j-1} h_i \right]\!\right], \text{ and} \\ G_j &= \widehat{h_{j+1}} + \left\lfloor \frac{1+h_j+\left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right]}{u} \right\rfloor. \end{aligned}$$

Then

$$\begin{aligned} H_{j+1} &= H_j - (u-1)G_j \\ &= \left(u\widehat{h_{j+1}} + h_j + \left[\!\left[\sum_{i=0}^{j-1} h_i \right]\!\right]\right) - (u-1)\cdot \left\lfloor \frac{1+h_j+\left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right]}{u} \right\rfloor. \end{aligned}$$

Thus to prove that $H_{j+1} = \widehat{h_{j+1}} + \left[\!\left[\sum_{i=0}^{j} h_i\right]\!\right]$, the following needs to be shown:

$$\left[\!\left[\sum_{i=0}^{j} h_i\right]\!\right] = h_j + \left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right] - (u-1)\cdot\left\lfloor \frac{1+h_j+\left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right]}{u}\right\rfloor \tag{4.10}$$

for $j < \ell$, and $h_j, h_i \in \mathbb{Z}_u$. Note that $0 < \left(1+h_j+\left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right]\right) < 2u-1$. In order to show the veracity of equation (4.10), consider the following three cases.

Case 1: $\left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right] = 0$. Then the left hand side of equation (4.10) is

$$\begin{aligned}\left[\!\left[\sum_{i=0}^{j} h_i\right]\!\right] &= \left[\!\left[h_j + \sum_{i=0}^{j-1} h_i\right]\!\right] \\ &= \left[\!\left[h_j + \left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right]\right]\!\right] \\ &= [\![h_j]\!].\end{aligned}$$

The right hand side of equation (4.10) is $h_j - (u-1)\left\lfloor \frac{1+h_j}{u}\right\rfloor$. If $h_j < (u-1)$ then $h_j - (u-1)\left\lfloor \frac{1+h_j}{u}\right\rfloor = h_j = [\![h_j]\!]$ and if $h_j = (u-1)$ then $h_j - (u-1)\left\lfloor \frac{1+h_j}{u}\right\rfloor = (u-1)-(u-1) = 0 = [\![h_j]\!]$. Thus equation (4.10) holds for $\left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right] = 0$.

Case 2: $\left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right] \neq 0$ and $\left(1+h_j+\left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right]\right) < u$. Then $\left(h_j+\left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right]\right) < (u-1)$ and the left hand side of equation (4.10) is

$$\begin{aligned}\left[\!\left[\sum_{i=0}^{j} h_i\right]\!\right] &= \left[\!\left[h_j + \sum_{i=0}^{j-1} h_i\right]\!\right] \\ &= h_j + \left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right].\end{aligned}$$

As the right hand side of equation (4.10) is also equal to $h_j + \left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right]$, equation (4.10) holds in this case.

Case 3: $\left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right] \neq 0$ and $\left(1+h_j+\left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right]\right) \geqslant u$. Then $0 \leqslant \left(h_j-(u-1)+\left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right]\right) < (u-1)$ and the left hand side of equation (4.10) is

$$\begin{aligned}\left[\!\left[\sum_{i=0}^{j} h_i\right]\!\right] &= \left[\!\left[h_j - (u-1) + \sum_{i=0}^{j-1} h_i\right]\!\right] \\ &= h_j - (u-1) + \left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right].\end{aligned}$$

As $u \leqslant \left(1 + h_j + \left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right]\right) < 2u - 1$, then $\left\lfloor \frac{1+h_j+\left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right]}{u} \right\rfloor = 1$. Thus the right hand side of equation (4.10) is $h_j + \left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right] - (u-1)$, which is equal to the left hand side of equation (4.10).

These three cases cover all possibilities for equation (4.10). Therefore, this equation holds. Thus $H_{j+1} = \widehat{h_{j+1}} + \left[\!\left[\sum_{i=0}^{j} h_i\right]\!\right]$, which completes the proof for $H_j = \widehat{h_j} + \left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right]$. For the sequence G_j, it follows that:

$$\begin{aligned} G_j &= \left\lfloor \frac{1 + H_{j+1}}{u} \right\rfloor \\ &= \left\lfloor \frac{1 + \widehat{h_{j+1}} + \left[\!\left[\sum_{i=0}^{j} h_i\right]\!\right]}{u} \right\rfloor \\ &= \left\lfloor \frac{1 + u\widehat{h_{j+2}} + h_{j+1} + \left[\!\left[\sum_{i=0}^{j} h_i\right]\!\right]}{u} \right\rfloor \\ &= \widehat{h_{j+2}} + \left\lfloor \frac{1 + h_{j+1} + \left[\!\left[\sum_{i=0}^{j} h_i\right]\!\right]}{u} \right\rfloor. \end{aligned}$$

This concludes the proof for the sequence G_j. ■

The semi-closed form for G_j in the binary case $u = 2$ was claimed in [LAN02] without a proof. Fortunately, their claim follows straightforwardly from Theorem 4.11. Furthermore, note that in [LAN02], the upper limit of the summation in the decomposition of $\sigma \in [0, H]$ is wrongly set to $\lfloor \log_2 H \rfloor$ instead of $(\lfloor \log_2 H \rfloor - 1)$.

Corollary 4.12 (Binary case, [LAN02])
If $u = 2$ *then* $G_j = \widehat{h_{j+1}} + \left\lfloor \frac{1+h_j}{u} \right\rfloor = \left\lfloor \frac{H+2^j}{2^{j+1}} \right\rfloor$.

Proof
Theorem 4.11 shows that $G_j = \widehat{h_{j+1}} + \left\lfloor \frac{1+h_j+\left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right]}{u} \right\rfloor$. However, $u = 2$ implies that $\left[\!\left[\sum_{i=0}^{j-1} h_i\right]\!\right] = 0$. Thus $G_j = \widehat{h_{j+1}} + \left\lfloor \frac{1+h_j}{u} \right\rfloor$. By definition, the following holds: $\widehat{h_{j+1}} = \left\lfloor \frac{H}{u^{j+1}} \right\rfloor = \left\lfloor \frac{H}{2^{j+1}} \right\rfloor$. Thus

$G_j = \left\lfloor \frac{H}{2^{j+1}} \right\rfloor + \left\lfloor \frac{1+h_j}{u} \right\rfloor$. As $\widehat{h_j} = u\widehat{h_{j+1}} + h_j$, the following hold:

$$\begin{aligned} h_j &= \widehat{h_j} - u\widehat{h_{j+1}} \\ &= \left\lfloor \frac{H}{u^j} \right\rfloor - u \left\lfloor \frac{H}{u^{j+1}} \right\rfloor \\ &= \left\lfloor \frac{H}{2^j} \right\rfloor - 2 \left\lfloor \frac{H}{2^{j+1}} \right\rfloor ; \end{aligned}$$

$$\begin{aligned} \Longrightarrow G_j &= \left\lfloor \frac{H}{2^{j+1}} \right\rfloor + \left\lfloor \frac{1 + \left\lfloor \frac{H}{2^j} \right\rfloor}{2} \right\rfloor - \left\lfloor \frac{H}{2^{j+1}} \right\rfloor \\ &= \left\lfloor \frac{1}{2} \left\lfloor \frac{H+2^j}{2^j} \right\rfloor \right\rfloor \\ &= \left\lfloor \frac{H+2^j}{2^{j+1}} \right\rfloor . \end{aligned}$$

■

4.5 Sumset Based Range Proofs

The results of Theorem 4.10 allow for a more efficient range proof to be built, based on sumset decomposition. The efficiency gain appears when handling arbitrary ranges, as it would require the use of a single range proof instead of the AND composition with two range proofs. The general range proof problem $\sigma \in [A, B]$ is reduced to solving the range proof $(\sigma - A) \in [0, H]$, with $H = B - A$. Thus, the rest of this section focuses on the range proof problem $\sigma \in [0, H]$. Theorem 4.10 shows that $\sigma \in [0, H]$ can be decomposed as $\sigma = \omega' + \sum_{j=0}^{\ell(u,H)-1} \sigma_j G_j$, with $\sigma_j \in \mathbb{Z}_u$ and $\omega' \in [0, H']$. Recall that G_j are public elements. Also recall that Theorem 4.10 states that if $(u-1) \mid H$ then $H' = 0$ and there is no element ω'. In the latter, proving that all of the elements σ_j are in $\mathbb{Z}_u$ can be achieved efficiently with the same set membership proof, exactly as in Protocol 4.1 or 4.4. However, the existence of $\omega' \in [0, H']$ imposes the need to choose between two solutions:

1. either an additional set membership proof for ω' needs to be added to the protocol,
2. or the range $[0, H]$ needs to be artificially increased,
   to obtain a proof of the form $\sigma(u-1) \in [0, H(u-1)]$.

The modification achieved in the second option takes advantage of the property brought by $(u-1) \mid H(u-1)$, which is the suppression of the element ω' as $H' = 0$. Furthermore, when using either Protocol 4.1 or 4.4, multiplying the range by $(u-1)$ implies that ℓ will be increased by 1, as $H \leqslant u^L$ implies that $H(u-1) < u^{L+1}$, and thus $\ell \leqslant L+1$. Increasing ℓ by 1 implies an additional communication of 4 elements $\left(V_\ell, (a_\ell, \text{ or } E_\ell), z_{\sigma_\ell}, z_{v_\ell}\right)$. This additional communication load is identical to the cost of an additional set membership proof regarding

the basic communication load. However, in the case of the first solution for ω', where an additional set membership proof is needed, the setup cost will increase. This increase amounts to H' additional elements that correspond to the signatures of the elements in $\mathbb{Z}_{H'}$, and to the public key of the signature scheme. Therefore, the second solution, with the artificial increase of the range $[0, H]$, is the most efficient solution.

An alternative solution suggested in [CLs10], is to use the OR-proof instead of the additional set membership proof, to prove that ω' is one of the elements in $[0, H']$. However, as OR-proofs have linear complexity, they are less efficient for individual elements when compared to set membership proofs. Furthermore, their security is restricted to witness indistinguishability (see Section 2.3.5), therefore OR-proofs should be discarded as a solution for this specific task.

Note that the range proof enhancement techniques of [CLs10], that are explained in this section, apply to any range proof based on range decomposition. Moreover, the range proof presented in [CLs10] is built upon Protocol 4.1. The solution presented here is built upon Protocol 4.4, as it is slightly more efficient. The aim is to prove that for a commitment C, C^{u-1} commits to a value in $[0,(u-1)H]$ by using Protocol 4.4. As $(u-1) \mid H(u-1)$, Theorem 4.10 states that $H' = 0$ and thus $\sigma(u-1) = \sum_{j=0}^{\ell(u,H(u-1))-1} \sigma_j G_j$.

Computational assumptions. The computational hardness assumptions required for the range proof in this section are identical to those for Protocol 4.4, namely the u-Strong Diffie-Hellman assumption as well as bilinear groups (see Section 2.1.3) and their associated computational hardness assumptions. Note that here again, Pedersen commitments are chosen as the commitment scheme used.

Protocol explanation. The sumset based range proof is presented in Protocol 4.5. As it is based on Protocol 4.4, the necessary modifications to change the u-ary decomposition to a sumset decomposition are highlighted by boxing the elements that differ. Recall that the elements G_j are computed either recursively as in equation (4.6), with $H_0 = H(u-1)$, or with the semi-closed form as described in Theorem 4.11, where $\widehat{h_j} = \left\lfloor \frac{H(u-1)}{u^j} \right\rfloor$ and $H(u-1) = \sum_{j=0}^{L-1} h_j u^j$. The sumset decomposition will define the value $\ell \leqslant L+1$ for $H \leqslant u^L$. The decomposition of $\sigma(u-1)$ will require to prove the set membership of ℓ elements $\sigma_j \in \mathbb{Z}_u$, as $\sigma(u-1) = \sum_{j=0}^{\ell-1} \sigma_j G_j$.

For the underlying proof of knowledge, recall that raising the commitment to the power $(u-1)$ is equivalent to multiplying the secret σ by $(u-1)$. As the elements σ_j are all in $\mathbb{Z}_u$, Protocol 4.4 remains the same regarding the set membership proofs $\sigma_j \in \mathbb{Z}_u$. However, their composition towards $C^{(u-1)}$ has to be changed. Thus the computation of D needs to be modified as $D = h^{m(u-1)} g^{\sum_{j=0}^{\ell-1}(s_j G_j)}$ and the verification check on D becomes:

$$D \stackrel{?}{=} C^{c(u-1)} h^{z_r(u-1)} g^{\sum_{j=0}^{\ell-1}\left(z_{\sigma_j} G_j\right)}.$$

Common Input: $g, h, u, \ell \leftarrow \boxed{\ell(u, H(u-1))}$,
$\boxed{\{G_j\}_{j\in\mathbb{Z}_\ell}}$, and a commitment C.

Prover Input: σ, r such that $C = g^\sigma h^r$ and $\sigma \in [0, H]$.

$P \xleftarrow{y,\{A_i\}} V$
- Verifier picks $x \in_R \mathbb{Z}_p^*$ such that $-x \notin \mathbb{Z}_u$ and sends $y \leftarrow g^x$ and $A_i \leftarrow g^{\frac{1}{x+i}}$, $\forall i \in \mathbb{Z}_u$.
- Prover checks that $y \in \mathbb{G}_1^*$, $A_i \in \mathbb{G}_1^*$ and that $e(g,g) \stackrel{?}{=} e(A_i,\ y \cdot g^i)$, $\forall i \in \mathbb{Z}_u$.

$P \xrightarrow{\{V_j\}} V$
- Prover picks $v_j \in_R \mathbb{Z}_p^*$ and sends $V_j \leftarrow A_{\sigma_j}^{v_j}$, for every $j \in \mathbb{Z}_\ell$, such that $\boxed{\sigma(u-1) = \sum_{j=0}^{\ell-1}\left(\sigma_j G_j\right)}$.
- Verifier checks that $V_j \in \mathbb{G}_1^*$, $\forall j \in \mathbb{Z}_\ell$.

Prover and Verifier run

$$\mathrm{PK}\left\{\left(\{\sigma_j\}, r, \{v_j\}\right) : \boxed{C^{(u-1)} = h^{r(u-1)} g^{\sum_{j=0}^{\ell-1}(\sigma_j G_j)}} \ \wedge\ V_j = g^{\frac{v_j}{x+\sigma_j}},\ \forall j \in \mathbb{Z}_\ell\right\}$$

$P \xrightarrow{\{E_j\},D} V$
- Prover picks $s_j, t_j, m \in_R \mathbb{Z}_p^*$ for every $j \in \mathbb{Z}_\ell$ and sends $E_j \leftarrow V_j^{-s_j} g^{t_j}$ and $\boxed{D \leftarrow h^{m(u-1)} g^{\sum_{j=0}^{\ell-1}(s_j G_j)}}$.
- Verifier checks that $E_j, D \in \mathbb{G}_1^*$, $\forall j \in \mathbb{Z}_\ell$.

$P \xleftarrow{c} V$
- Verifier sends a random challenge $c \in_R \mathbb{Z}_p^*$.
- Prover checks that $c \in \mathbb{Z}_p^*$.

$P \xrightarrow{\{z_{\sigma_j}\},\{z_{v_j}\},z_r} V$
- Prover sends $z_r \leftarrow (m - rc)$, and $z_{\sigma_j} \leftarrow (s_j - \sigma_j c)$, $z_{v_j} \leftarrow (t_j - v_j c)$ for every $j \in \mathbb{Z}_\ell$.
- Verifier checks that $z_{\sigma_j}, z_{v_j}, z_r \in \mathbb{Z}_p^*$ for every $j \in \mathbb{Z}_\ell$, that $\boxed{D \stackrel{?}{=} C^{c(u-1)} h^{z_r(u-1)} g^{\sum_{j=0}^{\ell-1}\left(z_{\sigma_j} G_j\right)}}$ and that $E_j \stackrel{?}{=} V_j^{cx - z_{\sigma_j}} g^{z_{v_j}}$ for every $j \in \mathbb{Z}_\ell$.

Protocol 4.5 – Interactive range proof protocol for range $[0, H]$

Theorem 4.13

If the u-Strong Diffie-Hellman assumption associated with a pairing generator PG *holds, then Protocol 4.5 is a zero-knowledge range argument for the range* $[0, H]$.

Proof

The proof to show that Protocol 4.5 is a zero-knowledge range argument, is identical to the proof of Theorem 4.6 with minor modifications. *Completeness* is achieved as Theorem 4.10 ensures that $\sigma(u-1) = \sum_{j=0}^{\ell-1} \sigma_j G_j$. *Special soundness* is achieved similarly as in Theorem 4.6, with the difference that σ is extracted with the following formula:

$$\sigma = \frac{\sum_{j=0}^{\ell-1} \left(z_{\sigma_j} - z'_{\sigma_j} \right) G_j}{c' - c}.$$

The simulator for the *special honest verifier zero-knowledge* is achieved as described in Fig-

1. Sim retrieves $y, \{A_i\}$ from V^* (or from a trusted third party).
2. Sim chooses $\sigma \in_R [0, H]$, $v_j \in_R \mathbb{Z}_p^*$ for every $j \in \mathbb{Z}_\ell$, and computes $V_j \leftarrow A_{\sigma_j}^{v_j}$ where $\sigma(u-1) = \sum_{j=0}^{\ell-1} \sigma_j G_j$.
3. Sim runs the simulator of
$$\mathrm{PK}\left\{ \left(\{\sigma_j\}, r, \{v_j\}\right) : C^{(u-1)} = h^{r(u-1)} g^{\sum_{j=0}^{\ell-1}(\sigma_j G_j)} \ \wedge \ V_j = g^{\frac{v_j}{x+\sigma_j}}, \ \forall j \in \mathbb{Z}_\ell \right\} :$$
   (a) On challenge $c \in \mathbb{Z}_p^*$, Sim chooses $\{z_{\sigma_j}\}, \{z_{v_j}\}, z_r \in_R \mathbb{Z}_p^*$.
   (b) Finally, Sim computes $D \leftarrow C^{c(u-1)} h^{z_r(u-1)} g^{\sum_{j=0}^{\ell-1}\left(z_{\sigma_j} G_j\right)}$ and $E_j \leftarrow V_j^{-z_{\sigma_j} - \sigma_j c} g^{z_{v_j} + v_j c}$ for every $j \in \mathbb{Z}_\ell$.
4. Sim returns the transcript $\left\{y, \{A_i\}, \{V_j\}, \{E_j\}, D, c, \{z_{\sigma_j}\}, \{z_{v_j}\}, z_r\right\}$.

Figure 4.5 – Simulator for the interactive range argument protocol, based on the sumset decomposition

ure 4.5. Note that it is essentially the same simulator as for Theorem 4.6, with the appropriate modifications for the sumset decomposition. ■

Communication and Computational Complexity. The communication load and computational complexity formulas are identical to the ones of Protocol 4.4. Moreover, the computation of u is achieved in the same way as in Protocol 4.4. However, the parameter ℓ is computed as $\ell = \ell(u, H(u-1)) \leqslant L+1$ for $H \leqslant u^L$. This implies that ℓ is increased by one when compared to Protocol 4.4. Furthermore, the complexities in this case apply for arbitrary ranges, which makes them roughly twice as efficient. Thus, for the range $[A, B]$ with $H = B - A$, the communication load is

$$Com(u, \ell) = 2\ell \cdot \left(\|\mathbb{G}_1\| + \left\|\mathbb{Z}_p\right\| \right) + \left(\|\mathbb{G}_1\| + 2 \cdot \left\|\mathbb{Z}_p\right\| \right).$$

The computational complexity of the prover amounts to $(3\ell+2)$ exponentiations, and the one of the verifier consists of $(2\ell+3)$ exponentiations.

Concrete Example and Comparisons The performance of the different methods depends on the application at hand as well as the assumptions one is willing to make. Assume, at first, that all possible assumptions are acceptable. For intervals containing 7 elements or less, employing the set membership proof presented in Section 3.3 directly would be more efficient. Beyond 7, elements and for ranges smaller than 122 bits, the sumset based range proof presented in Protocol 4.5 outperforms all other existing range proofs. For ranges that are larger than 122 bits, the square decomposition method by Lipmaa in [Lip03] and by Groth in [Gro05] are favorable as they are mostly independent of the size of the interval. However, the prover will need to run the Rabin-Shallit algorithm (or an equivalent algorithm) to represent numbers as the sum of four squares (three in the case of [Gro05]) and this kind of algorithm has a quadratic complexity in the bit-length of the number to be decomposed. This means that for a n bit-length number, the complexity will be $O(n^2)$.

Note that the different existing protocols have different security goals. In order to provide meaningful comparisons, one has to set a unique security objective and perform the complexity computations accordingly. The security goal is set to 128 bit security for this thesis, meaning that a cheating prover will succeed with a soundness probability of at most 2^{-128}. For the protocols presented in this chapter, this security goal implies that the security parameter k has to be $k \geqslant 256 + \log_2 u$. Recall that the security parameter defines the size of the group $\mathbb{G}_1$ ($\|\mathbb{G}_1\| = k$). In the following, it is considered that $\|\mathbb{G}_1\| = 256$ to ease comparisons with other protocols. Nevertheless, when used in practice, the size of $\mathbb{G}_1$ should be $\|\mathbb{G}_1\| = 256 + \log_2 u$. Previous range proofs were often defined with a 80 bit security goal, therefore their complexities become significantly higher with the 128 bit security objective.

As the protocols in this chapter require bilinear pairings, it is important to recall that the sizes of $\mathbb{G}_1$ and $\mathbb{G}_T$ will depend on the security objectives. Galbraith, Paterson, and Smart provide a detailed explanation in [GPS08] on which size to use for $\mathbb{G}_1$ and $\mathbb{G}_T$ according to different security settings and requirements. Two different recommendations are specified here. For a 128 bit security, NIST [NIST12] recommends to use $\|\mathbb{G}_1\| = 256$ bits and $\|\mathbb{G}_T\| = 3072$ bits. However, Lenstra [Len06] recommends the use of $\|\mathbb{G}_1\| = 256$ bits and $\|\mathbb{G}_T\| = 4440$ bits.

Before explaining the details of the comparisons, a concrete example is provided. Assume that a bank wants to provide special offers from a third party to its young clients. However the exact age of clients should not be divulged to the third party. This offer targets those who are born between 1990 and 1998 (not included). The conversion of the birth date into the Unix Epoch system, results in a target range of $[631152000, 883612800]$. Figure 4.6 provides a comparison amongst the relevant protocols, ordered by communication complexity. Figure C.1 in Appendix C provides a comparison regarding computational complexity, with the same order as in Figure 4.6.

Scheme	*Communication Complexity*
CCs_AND_Lenstra (Protocol 4.2 with [Len06] recommendations)	63264 bits
CCs_AND_NIST (Protocol 4.2 with [NIST12] recommendations)	54528 bits
Lipmaa [Lip03] (Sum of 4 squares)	36352 bits
Boudot [Bou00] (Square + CFT [CFT98b])	32294 bits
Groth [Gro05] (Sum of 3 squares)	29440 bits
Scemama [Sce09] (Square + CFT [CFT98b])	28668 bits
CCs_AND_Arfaoui (AND composition with Protocol 4.4)	19200 bits
Groth [Gro11] (binary decomposition of commitments of commitments)	12032 bits
Sumset based range proof (Protocol 4.5)	11008 bits

Figure 4.6 – Communication load comparison for range proof [631152000, 883612800]

The first schemes discussed, are the ones of Lipmaa [Lip03] and Groth [Gro05]. Both of them are based on the sum of squares, and were initially defined for 80 bit security. Moreover, they are focused on positivity testing, which means that two range proofs are required in order to handle arbitrary ranges. The communication load equation is solely provided here, with the values of the parameters. The details of those parameters can be found in [Lip03]:

$$Com_{[\text{Lip03}]} = 2 \cdot \left(6\|\mathcal{G}\| + 14k + 5B + 10\|F(k)\| + \frac{5}{2}\|M\|\right),$$
$$Com_{[\text{Gro05}]} = 2 \cdot (5\|\mathcal{G}\| + 11k + 4B + 8\|F(k)\| + 2\|M\|),$$

where $\|M\| = B = \|\mathcal{G}\| = 1024$, $k = 128$, and $\|F(k)\| = 256$.

The schemes of Boudot [Bou00] and Scemama [Sce09] are both based on the conjunction of a square proof with the Chan et al. method [CFT98b]. Here again, these schemes were designed with 80 bit security. The communication load equation is solely provided here, with the values of the parameters. The details of those parameters can be found in [Bou00]:

$$Com_{[\text{Bou00}]} = 14 + 7s + 58t + 18\|n\| + 5\|b\| + 7\|b-a\|,$$
$$Com_{[\text{Sce09}]} = 12 + 6s + 50t + 16\|n\| + 5\|b\| + 6\|b-a\|,$$

where $s = 40$, $t = 128$, $\|n\| = 1024$, and $\|b\| = \|b-a\| = 512$. Note that the upper bound of the range has to be lower than 512 bits for both of these schemes.

The second scheme of Groth [Gro11] achieves a weaker security than existing range proofs. Moreover, it requires a non-intuitive description with a binary decomposition of commitments of commitments. Due to the difficulty of understanding the protocol itself, several mistakes are present in the original paper. A preliminary corrected version has been privately disclosed to the author of this thesis. Although the security proof is still difficult to analyse, a corrected version should be publicly released soon. The communication load achieved by the protocol in the corrected version of [Gro11], amounts to $17+10\left(\log_2(B-A)\right)^{1/3}$ group elements.

The CCs_AND_Lenstra and the CCs_AND_NIST schemes correspond to Protocol 4.2 with the recommendations for group size from Lenstra [Len06] and from NIST [NIST12] respectively. Recall that the communication load of Protocol 4.2 is:

$$Com_{\text{Protocol 4.2}}(u,\ell) = 2\ell\cdot\left(\|\mathbb{G}_1\|+\|\mathbb{G}_T\|+2\cdot\left\|\mathbb{Z}_p\right\|\right)+\left(\|\mathbb{G}_1\|+2\cdot\left\|\mathbb{Z}_p\right\|\right).$$

The values of $\left\|\mathbb{Z}_p\right\|$, $\|\mathbb{G}_1\|$, and $\|\mathbb{G}_T\|$ depend on the recommendations used. Computing u is achieved by solving equation (4.5) (Section 4.3), which will determine ℓ as $u^\ell > (B-A)$. Note that equation (4.5) depends on the values of $\left\|\mathbb{Z}_p\right\|$, $\|\mathbb{G}_1\|$, and $\|\mathbb{G}_T\|$, thus different recommendations will lead to different values for u and ℓ. The following communication load equations are therefore obtained:

$$\begin{aligned} Com_{\text{CCs\_AND\_Lenstra}}(26,6) &= 2\cdot 6\cdot(256+4440+2\cdot 256)+(256+2\cdot 256)\\ &= 63264 \text{ bits},\\ Com_{\text{CCs\_AND\_NIST}}(22,7) &= 2\cdot 7\cdot(256+3072+2\cdot 256)+(256+2\cdot 256)\\ &= 54528 \text{ bits}. \end{aligned}$$

Considering the communication load of previous range proofs with similar security requirements, CCs_AND_Lenstra becomes more efficient when $\ell \leqslant 2$. CCs_AND_NIST becomes more efficient when $\ell \leqslant 3$. If the computation of u is restricted with equation (4.5), this leads to ranges that are smaller in size than 169 in the case of CCs_AND_Lenstra, and ranges that are smaller in size than 2744 in the case of CCs_AND_NIST. These limits are obtained by solving the following equations[4]:

$$\ell \leqslant \frac{28668-\left(\|\mathbb{G}_1\|+2\cdot\left\|\mathbb{Z}_p\right\|\right)}{2\cdot\left(\|\mathbb{G}_1\|+\|\mathbb{G}_T\|+2\cdot\left\|\mathbb{Z}_p\right\|\right)}, \text{ and}$$

$$u\log u \leqslant \ell\cdot\frac{\left(\|\mathbb{G}_1\|+\|\mathbb{G}_T\|+2\cdot\left\|\mathbb{Z}_p\right\|\right)}{\|\mathbb{G}_1\|}.$$

[4] The first equation is obtained from $Com_{\text{Protocol 4.2}}(u,\ell) \leqslant Com_{[\text{Sce09}]}$.

The CCs_AND_Arfaoui scheme corresponds to the AND composition applied to Protocol 4.4. The principal advantage of this scheme is that it replaces the elements from $\mathbb{G}_T$ with elements from $\mathbb{G}_1$. Thus the communication load becomes:

$$Com_{\text{CCs\_AND\_Arfaoui}}(u,\ell) = 4\ell \cdot \left(\|\mathbb{G}_1\| + \left\|\mathbb{Z}_p\right\|\right) + \left(\|\mathbb{G}_1\| + 2 \cdot \left\|\mathbb{Z}_p\right\|\right).$$

As no elements from $\mathbb{G}_T$ are transmitted, the more restrictive recommendations from Lenstra [Len06] will only impact the computational complexity of the prover during the setup phase. Finding the value of u will also be achieved by solving equation (4.5), however the value of c_2 is changed to $c_2 = 2 \cdot \|\mathbb{G}_1\| + 2 \cdot \left\|\mathbb{Z}_p\right\|$. Therefore, equation (4.5) becomes $u\left(\log u\right)^2 = 4\log\left(B+1-A\right)$. This implies that $u = 11$ which in turn implies that $\ell = 9$. Thus, the communication load sums up to $Com_{\text{CCs\_AND\_Arfaoui}}(11,9) = 19200$ bits. Compared to previous range proofs with similar security requirements, the CCs_AND_Arfaoui scheme is more efficient for $\ell \leqslant 13$. This is obtained by solving:

$$\ell \leqslant \frac{28668 - \left(\|\mathbb{G}_1\| + 2 \cdot \left\|\mathbb{Z}_p\right\|\right)}{4 \cdot \left(\|\mathbb{G}_1\| + \left\|\mathbb{Z}_p\right\|\right)}.$$

The restriction $(B-A) < u^\ell$ implies that $u\left(\log u\right) \leqslant 4\ell$. Therefore, the CCs_AND_Arfaoui scheme is more efficient for ranges smaller than 50 bits.

Last but not least, the communication load of the sumset based range proof presented in Protocol 4.5, is obtained by:

$$Com_{\text{Protocol 4.5}}(u,\ell) = 2\ell \cdot \left(\|\mathbb{G}_1\| + \left\|\mathbb{Z}_p\right\|\right) + \left(\|\mathbb{G}_1\| + 2 \cdot \left\|\mathbb{Z}_p\right\|\right).$$

Note that the computation of u is identical to the one of the CCs_AND_Arfaoui scheme. However, the definition of ℓ differs from the previous cases. The value of ℓ for the sumset based range proof is equal to the ℓ for the CCs_AND_Arfaoui scheme plus one. Therefore, $Com_{\text{Protocol 4.5}}(11,10) = 11008$ bits. Compared to previous range proofs with similar security requirements, the sumset based range proof is more efficient for $\ell \leqslant 26$. This is obtained by solving:

$$(\ell+1) \leqslant \frac{28668 - \left(\|\mathbb{G}_1\| + 2 \cdot \left\|\mathbb{Z}_p\right\|\right)}{2 \cdot \left(\|\mathbb{G}_1\| + \left\|\mathbb{Z}_p\right\|\right)}.$$

The restriction $(u-1)H < u^{\ell+1}$ implies that $u\left(\log u\right) \leqslant 4\ell$. Therefore, the sumset based range proof is more efficient for ranges smaller than 122 bits. Above this limit, the sum of squares method becomes more efficient.

In the contentious case of [Gro11], their range proof is supposed to be more efficient for ranges between 39 and 941 bits. Bellow 39 bits, the sumset based range proof remains more efficient, and above 941 bits, the sum of squares method from [Gro05] becomes more efficient. The lower bound limit can be derived by assuming that $4\ell = u(\log_2 u)$, that $4\log_2(B-A) = u(\log_2 u)^2$, and by solving:

$$0 \leqslant \left(17 + 10\left(\log_2(B-A)\right)^{1/3}\right) - (4(\ell+1)+3).$$

The upper bound is found by solving: $29440 \geqslant \left(17 + 10\left(\log_2(B-A)\right)^{1/3}\right)\|\mathbb{G}_1\|$.

Chapter 5

Non-Interactive Range Proofs, Without Random Oracles

This chapter starts by presenting the non-interactive range proof primitive in Section 5.1. Section 5.2 then describes prior and related work, as well as the recent results of Fauzi, Lipmaa, and Zhang [FLZ13], and Lipmaa [Lip14b, Lip16]. The difficulties in creating a non-interactive range proof will be illustrated in Section 5.3, by showing that the non-interactive range proof of Yuen et al. [YHM+09] is insecure. Section 5.4 will explain a non-interactive subargument that is necessary for the non-interactive range proof presented in Section 5.5. The main results of sections 5.3, 5.4, and 5.5 are published in the proceedings of FC 2012 [CLZ12] as a joint work with Helger Lipmaa and Bingsheng Zhang.

5.1 Non-Interactive Range Proofs Primitive

In this chapter, the problem being solved is an extension of the range proof explained in the previous chapter. This extension restricts the type of interactions between the prover and the verifier. In the last chapter, provers and verifiers were allowed to interact with each other, meaning that they could reply to one another until the necessary messages were exchanged to complete the range proof. In this chapter, the interactions are suppressed and communications are reduced to a single message sent by the prover to any verifiers. The focus in this chapter is, therefore, to provide non-interactive zero-knowledge (NIZK) range arguments to solve the non-interactive range proof primitive. The straightforward solution consists of using a generic transformation of the interactive protocols in order to obtain their non-interactive counterpart in the random oracle model (see Section 2.3.6). The goal of this chapter is to provide an alternative without the random oracle model. The security will thus be proven in the common reference string model (see Section 2.2.4) instead of the random oracle model. Users will thus have the possibility of selecting protocols according to which security model suits them best, as explained in Section 2.3.6.

In addition to the usual interest for range proofs (as explained in Section 4.1), the non-interaction property is also gaining popularity. Indeed, an increasing number of applications require some statements to be verified multiple times by different parties, provided that these verifiers are disallowed to interact with provers. Thus, the non-interaction property solves the problem by sending (or broadcasting) a single message containing the entire proof of a statement to be proven. Concrete examples of applications that require non-interactive range proofs include *e-voting* and *e-auctions*. Non-interactive range proofs are also starting to be considered as cryptographic building blocks, as they are included in new primitives such as *graded signatures*.

In the case of *e-voting*, the validity of ballots needs to be verified. Not only does this verification need to be performed by the tallying server when votes are being cast, verification also needs to be possible by third parties for election monitoring purposes. However, these third parties do not have access to voters. Instead they receive validity proofs as single non-interactive messages. Therefore, non-interactive range proofs are essential for election monitoring in e-voting. A similar requirement is present in *e-auctions*. For instance, in the case of proxy auctions, the auction server manages bids, updates the current price of items to the winning bid, and keeps the maximum bid of the winning client private. However, the trust of users in the auction server is questionable as it could artificially set the price to the maximum bid of the winning client. Hence, clients need to check the correctness of all of the updates performed on the price of the item, as well as the current price, without being able to contact the other bidders. This is achieved by transmitting non-interactive proofs to the clients, as demonstrated in [CHS04]. The new primitive called *graded signatures* was introduced recently by Osmanoglu in [Osm15]. In this primitive, the signature process of registered users is assimilated as a positive grade. Thus, the aim of the primitive is to collect and combine these signatures into a general graded signature, that conceals the identity of the signers. Osmanoglu shows in [Osm15] how to construct such a graded signature scheme from a non-interactive range proof.

Definition 5.1 (Non-Interactive Range Proof)
Let $C = (\mathrm{Gen}_{com}, \mathrm{Com}, \mathrm{Open})$ *be the generation, the commit, and the open algorithm of a string commitment scheme in the common reference string (CRS) model. A non-interactive range proof with respect to the commitment scheme* C *in the CRS model is a special case of the interactive range proof, where communications are reduced to a single message and a common reference string is assumed to be shared among users. Hence, for an instance* c*, a non-interactive range proof with respect to commitment scheme* C *and integer range* $[A,B]$ *is a non-interactive zero-knowledge proof of knowledge (NIZK-PK) for the following statement:*

$$\textit{NIZK-PK}\{(\sigma,\rho) : c \leftarrow \mathrm{Com}(\sigma;\ \rho) \wedge \sigma \in [A,B]\}\textit{, where } A,B \in \mathbb{N}.$$

Remark: As in the case of interactive range proofs, the proof system for non-interactive range proofs is defined for *any* commitment scheme. The statement being proven is the ability for the prover to open his commitment to an element contained in the public range $[A,B]$. It is

also important to note that non-interactive range proofs are often non-interactive arguments, for the same reasons as for interactive range proofs. Since the cryptographic literature (past, recent, and related) refers to the problem as a "non-interactive range proof", that term is used in this thesis. Some additional explanations are provided in the remarks that are in Section 4.1 and Section 3.1.

An alternative definition can be obtained by replacing the string commitment scheme with a public key cryptosystem.

Definition 5.2 (Non-Interactive Range Proof with respect to public key cryptosystems) *Let $\mathscr{E} = (\text{Gen}_{pkc}, \text{Enc}, \text{Dec})$ be the generation, the encryption, and the decryption algorithm of a public key cryptosystem. A non-interactive range proof with respect to the public key cryptosystem $\mathscr{E}$ is a special case of the non-interactive range proof with respect to the commitment scheme C in the CRS model. For an instance e, a non-interactive range proof with respect to the public key cryptosystem $\mathscr{E}$ and integer range $[A,B]$, is a non-interactive proof of knowledge for the following statement:*

$$\textit{NIZK-PK}\{(\sigma,\rho) : e \leftarrow \text{Enc}(\sigma;\rho) \wedge \sigma \in [A,B]\}, \textit{ where } A,B \in \mathbb{N}.$$

Remark: In Definition 5.2, the proof system for non-interactive range proofs is defined for *any* public key cryptosystem. The statement being proven is the ability for the prover to decrypt his ciphertext into an element contained in the public range $[A,B]$.

The common approach to construct non-interactive arguments is to apply known generic transformations to interactive arguments. However, generic transformations often require specific conditions and impose drawbacks. For instance, several generic transformations such as the Fiat-Shamir heuristic [FS86], require the security to be proven in the random oracle model.

A better, but more complex, approach is to directly construct the non-interactive argument with a security proof alongside it. Unfortunately, the difficulty of this approach can easily lead to insecure protocols. This is, for instance, the case with the protocol presented by Yuen et al. in [YHM+09]. The insecurity of their protocol is demonstrated in Section 5.3, where the main idea of the attack comes from using Pedersen commitments in a group of known order. As they rely on the Lagrange theorem to prove that a non-negative number is the sum of four squares, their protocol can only conclude that the sum of four squares is computed modulo the group order. Hence an attacker can prove that any number is "non-negative" and completely break the protocol in [YHM+09].

A correct solution is achieved in Section 5.5, where the NIZK range proof presented works in the common reference model for an encrypted secret σ, with respect to the lifted BBS public key cryptosystem [LZ12, BBS04] (see Section 2.4.4). Note that if σ needs to be committed, one can use the lifted BBS cryptosystem as a perfectly binding commitment. The construction of the NIZK range proof is achieved by using recent NIZK arguments by Groth and Lipmaa [Gro10,

Lip12a]. It also uses the additive combinatorics results from Section 4.4, that decompose the range proof $\sigma \in [0, H]$ based on the fact that $(u-1)\sigma \in (u-1)\cdot[0, H]$ if and only if $\sigma = \sum_{i=0}^{\ell-1} G_i \sigma_i$ and $\sigma_i \in [0, u-1]$, where G_i are as defined in Section 4.4. However, in contrast to Section 4.5, the proof that $\sigma_i \in \mathbb{Z}_u$ is done without the use of a signature scheme, but rather with a recursive use of the method in Section 4.4 and from [LAN02]. This recursive method shows that $\sigma_i = \sum_{j=0}^{\ell_v-1} G'_j \sigma'_{j,i}$ with $\sigma'_{j,i} \in [0,1]$. Here, $\ell_v := \lfloor \log_2(u-1) \rfloor$. By using the commitment scheme of [Gro10, Lip12a] that enables the succinct commitment to a vector $(\sigma_0, \ldots, \sigma_{\ell-1})$, and the Hadamard product argument of [Gro10, Lip12a] (see Section 2.4.7), all ℓ_v small range proofs can be done in parallel. The new range proof does not rely on the random oracle model nor use any proofs of knowledge of signatures. Furthermore, the NIZK range proof achieves sublinear communication and computational complexity by using the Groth-Lipmaa knowledge commitment [Gro10, Lip12a] (see Section 2.4.2) in conjunction with the lifted BBS cryptosystem [LZ12, BBS04] (see Section 2.4.4).

As a brief reminder, to commit to a vector $\boldsymbol{a} = (a_0, \ldots, a_{n-1})$, the Groth-Lipmaa knowledge commitment first takes the following as input: a common reference string crs_t, $\boldsymbol{a}$, and randomness r. It then outputs the commitment $\left(A, \widehat{A}\right)$ such that

$$\begin{aligned}\left(A, \widehat{A}\right) &= \mathrm{Com}^t(\mathsf{crs}_t; \boldsymbol{a}; r) \\ &= \left(g_t^r \prod_{i=0}^{n-1} g_{t,\lambda_i}^{a_i}, \quad \widehat{g}_t^r \prod_{i=0}^{n-1} \widehat{g}_{t,\lambda_i}^{a_i} \right),\end{aligned}$$

where g_{t,λ_i} and $\widehat{g}_{t,\lambda_i}$ are parameters contained in crs_t such that $\widehat{g}_{t,\lambda_i} = g_{t,\lambda_i}^{\widehat{\alpha}}$ for some secret $\widehat{\alpha}$, and $g_{t,\lambda_i} = g_t^{x^{\lambda_i}}$ for some secret x and where $\{\lambda_i\}_{i \in \mathbb{Z}_n} \subset \mathbb{N}$ is such that $\forall i < j\colon 0 < \lambda_i < \lambda_j$. Furthermore, $t \in \{1,2\}$ defines either a generator $g_1 \in \mathbb{G}_1$ or $g_2 \in \mathbb{G}_2$.

The lifted BBS cryptosystem [LZ12, BBS04] encrypts a message σ with randomness r_f and r_h with the following ciphertext:

$$\begin{aligned}\left(c_g, c_f, c_h\right) &= \mathrm{Enc}_{\mathsf{pk}}\left(\mathsf{crs}_1; \sigma; r_f, r_h\right) \\ &= \left(g_1^{r_f + r_h + \sigma}, f^{r_f}, h^{r_h}\right),\end{aligned}$$

where the secret key is $\mathsf{sk} = (\mathsf{sk}_1, \mathsf{sk}_2)$ and the public key is $\mathsf{pk} = \left(g_1, f, h\right) = \left(g_1, g_1^{1/\mathsf{sk}_1}, g_1^{1/\mathsf{sk}_2}\right)$. Further details can be found in Section 2.4.4.

The NIZK range proof described in this chapter requires a subargument that a knowledge-committed value is equal to a lifted BBS encrypted value. A novel solution for this subargument is described in Section 5.4, where the use of knowledge assumptions enables this subargument to be computationally more efficient than the one constructed by using Groth-Sahai proofs [GS08, GS12a], while keeping an identical communication load.

5.2 Prior, Recent, and Related Work

Although the interest in interactive range proofs started as early as 1987 with the work of Brickell et al. in [BCDvdG87], it is only in 2009 that the first NIZK range proof without random oracles was proposed by Di Crescenzo, Herranz, and Sáez in [CHS04]. However their solution is mainly of theoretical value. Their range proof targets statements of the form $\sigma > L$, where σ is decomposed in its binary form. Using the technique of Fischlin [Fis01] (although not cited), they reduce the complexity of their algorithm to the bit length of L. Let $n = \log_2 L$ be the bit length of L, then the proof system of [CHS04] requires at least n NIZK proofs of quadratic non residuosity from [SCP94].

The second attempt was undertaken by Yuen et al. in [YHM+09]. Their scheme uses the Lagrange theorem to decompose a positive number as the sum of four squares, similarly to the interactive version proposed by Lipmaa in [Lip03]. However the scheme in [YHM+09] uses Pedersen commitments with known group order, which render the scheme insecure as will be demonstrated in Section 5.3.

The range proof from Rial et al. [RKP09] combines the range proof of [CCs08] (described in Protocol 4.1 and explained in Section 4.3) with the Groth-Sahai NIZK proofs [GS08] and P-signatures [BCKL08]. However, the [RKP09] range proof is not claimed to be zero-knowledge, but only non-interactive witness indistinguishable (NIWI). It is nevertheless claimed in [RKP09] that NIZK should be achievable with the techniques from Groth-Sahai [GS08].

The range proof of Chaabouni, Lipmaa, and Zhang in [CLZ12], which is the focus of this chapter, opened the way for two more recent improvements [FLZ13, Lip14b, Lip16]. The improvement proposed by Fauzi, Lipmaa, and Zhang in [FLZ13] is to replace the permutation argument used in [CLZ12] by a shift argument. This replacement made it possible for them to obtain a protocol that is computationally more efficient with a slight communication efficiency improvement as well. Furthermore, the latest improvements brought by Lipmaa in [Lip14b, Lip16], enhance not only the shift argument, but also the product argument needed in [CLZ12] and in [FLZ13]. These enhancements allow for a major improvement in the computational complexity and in the communication load.

A related argument to the NIZK range proof is described in [FLZ14] with the construction of several NIZK set operations. The most notable set operation achieved is the pairwise multiset sum equality test (PMSET), where the prover aims to show that for four committed sets $\mathbb{A}_1, \mathbb{A}_2, \mathbb{A}_3$, and $\mathbb{A}_4$, all the elements contained in $\mathbb{A}_1$ and in $\mathbb{A}_2$ appear as many times as in $\mathbb{A}_3$ and in $\mathbb{A}_4$, regardless of their sorting or of their set membership proof. The PMSET argument then allows them to achieve other set operations such as the NIZK subset argument, the NIZK set intersection and union argument, or the NIZK set difference argument.

5.3 Breaking the COCOON 2009 NI Range Proof

In [YHM+09], Yuen et al. proposed a non-interactive range proof. This section shows that their argument is not secure.

Their goal was to prove that a committed secret σ is in some range $[A,B]$. To do so they prove that both $(\sigma - A)$ and $(B - \sigma)$ are non-negative by making use of the Lagrange theorem stating that any non-negative integer can be decomposed as the sum of four squares. Hence,

$$\sigma - A = \sum_{j=1}^{4} x_{1j}^2 \quad \text{and} \quad B - \sigma = \sum_{j=1}^{4} x_{2j}^2, \tag{5.1}$$

for some x_{ij}. The range proof of [YHM+09] is based on (symmetric) bilinear groups of composite order, that is, on bilinear groups $(n, \mathbb{G}_1, \mathbb{G}_T, e)$, where $n = pq$. To commit to a message σ, the committer picks a random[1] $r \in \mathbb{Z}_q$ and computes $C = g^\sigma h^r$, where g is a random generator of $\mathbb{G}_1$ (of order n), and h is a random generator of the subgroup $\mathbb{G}_q$ of $\mathbb{G}_1$ with order q. Given C, σ is uniquely determined in $\mathbb{Z}_p$, as $C^q = g^{\sigma q}$.

In their range proof, the prover finds the witnesses x_{ij} of equation (5.1) and outputs a proof

$$\pi = \left(\{C_{1j}, C_{2j}\}_{j \in \{1,2,3,4\}}, C, \varphi_1, \varphi_2 \right),$$

where

$$\begin{aligned}
C &\equiv g^\sigma h^r \in \mathbb{G}_1, \\
C_{ij} &\equiv g^{x_{ij}} h^{r_{ij}} \in \mathbb{G}_1 \text{ for } i \in \{1,2\} \text{ and } j \in \{1,2,3,4\}, \\
\varphi_1 &\equiv g^{-r+2\sum_{j=1}^4 r_{1j} x_{1j}} \cdot h^{\sum_{j=1}^4 r_{1j}^2} \in \mathbb{G}_1, \\
\varphi_2 &\equiv g^{r+2\sum_{j=1}^4 r_{2j} x_{2j}} \cdot h^{\sum_{j=1}^4 r_{2j}^2} \in \mathbb{G}_1.
\end{aligned}$$

The verifier checks if

$$e(h, \varphi_1) = e(g^A C^{-1}, g) \prod_{j=1}^{4} e(C_{1j}, C_{1j}), \text{ and} \tag{5.2}$$

$$e(h, \varphi_2) = e(Cg^{-B}, g) \prod_{j=1}^{4} e(C_{2j}, C_{2j}). \tag{5.3}$$

Now assume that a malicious prover $P^\star$ picks an integer $\sigma^* \in \{0, \ldots, p-1\} \setminus [A,B]$. This implies that either $(\sigma^* - A)$ or $(B - \sigma^*)$ is negative as an integer. Suppose $(B - \sigma^*) < 0$, then $P^\star$ chooses $\left\{x_{2j}^*\right\}_{j \in \{1,2,3,4\}}$ such that $n + (B - \sigma^*) = \sum_{j=1}^4 (x_{2j}^*)^2$. Then $P^\star$ sets $C \leftarrow g^{\sigma^*} h^r$, $C_{2j} \leftarrow g^{x_{2j}^*} h^{r_{2j}}$, φ_1 as above, and $\varphi_2 \leftarrow g^{r+2\cdot\sum_{j=1}^4 r_{2j} x_{2j}^*} \cdot h^{\sum_{j=1}^4 r_{2j}^2}$. Let $h = g^\alpha$ for some α. It is easy to see that the

[1] In [YHM+09], the scheme uses $r \in \mathbb{Z}_n$ to facilitate their security proof.

verification equation (5.3) still holds:

$$\begin{aligned} e(Cg^{-B},g)\prod_{j=1}^{4}e(C_{2j},C_{2j}) &= e(g,g)^{(\sigma^*-B)+\alpha r+\sum_{j=1}^{4}(x^*_{2j}+\alpha r_{2j})^2} \\ &= e(g,g)^{(\sigma^*-B)+\alpha r+\sum_{j=1}^{4}(x^*_{2j})^2+\sum_{j=1}^{4}\alpha^2 r^2_{2j}+2\sum_{j=1}^{4}\alpha r_{2j}x^*_{2j}} \\ &= e(g,g)^{\alpha\cdot\left(r+2\sum_{j=1}^{4}r_{2j}x^*_{2j}+\alpha\cdot\sum_{j=1}^{4}r^2_{2j}\right)} \\ &= e(h,\varphi_2). \end{aligned}$$

An identical construction can be made for the case that $(\sigma^* - A) < 0$, where the focus will be put on the verification equation (5.2). Therefore, it can be concluded that $P^\star$ is a polynomial time adversary who can always break the scheme. Hence, the NIZK range proof in [YHM+09] is not sound.

5.4 Equality Subargument of a lifted BBS Encryption and a Knowledge Commitment

The range proof of Section 5.5 requires a subargument that if $(A_c, \widehat{A}_c)$ is a knowledge-commitment of some σ (with $n = 1$ and some randomness $\widetilde{r} = r_c + r_f + r_h$), and (A_g, A_f, A_h) is a lifted BBS ciphertext of some σ' (with randomness $r = r_f + r_h$), then $\sigma = \sigma'$. That is, $A_c = g_1^{\widetilde{r}} g_{1,\lambda_0}^{\sigma} = g_1^{r_f+r_h+r_c} g_{1,\lambda_0}^{\sigma}$ and $(A_g, A_f, A_h) = (g_1^{r_f+r_h+\sigma}, f^{r_f}, h^{r_h})$ for randomness (r_c, r_f, r_h) and public key (f, h). The generator g_{1,λ_0} will be required in Section 5.5.

Computational assumptions. Beyond the need for asymmetric bilinear groups (see Section 2.1.3) and the associated computational hardness assumptions, the Groth-Lipmaa knowledge commitment additionally requires both the Λ-power symmetric discrete logarithm (Λ-PSDL) assumption and the Λ-power knowledge of exponent (Λ-PKE) assumption. Let PG_a be an asymmetric pairing group generator that on input 1^κ outputs descriptions of multiplicative cyclic groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ of prime order p where $\|p\| = \kappa$. Let $\mathbb{G}_1^* = \mathbb{G}_1 \setminus \{1\}$, $\mathbb{G}_2^* = \mathbb{G}_2 \setminus \{1\}$ and let $g_1 \in \mathbb{G}_1^*$, $g_2 \in \mathbb{G}_2^*$. The generated groups are such that there exists an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, meaning that:

- for all $a, b \in \mathbb{Z}_p$ it holds that $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$;
- $e(g_1, g_2) \neq 1$; and
- the bilinear map is efficiently computable.

One can implement an optimal asymmetric Ate pairing [HSV06] over a subclass of Barreto-Naehrig curves [BN05] very efficiently [GSNB11] (in that case, at a security level of 128-bits, an element of $\mathbb{G}_1/\mathbb{G}_2/\mathbb{G}_T$ can be represented in, respectively, 256/512/3072 bits).

The Λ-PSDL assumption illustrates the difficulty for an adversary to produce the secret element x from the set $\left\{g_1^{x^i},\ g_2^{x^i}\right\}_{i\in\{0\}\cup\Lambda}$, where $\Lambda=\{\lambda_i\}_{i\in\mathbb{Z}_n}\subset\mathbb{N}$ such that $\forall i<j\colon 0<\lambda_i<\lambda_j$. The Λ-PKE assumption regarding a bilinear group $\mathbb{G}_1$ for $t=1$ or $\mathbb{G}_2$ for $t=2$, states that given the set $\left\{g_t^{x^i},\ g_t^{\widehat{\alpha}x^i}\right\}_{i\in\{0\}\cup\Lambda}$ where $\widehat{\alpha}$ is secret, an adversary $\mathscr{A}$ can output a pair $(c,\widehat{c})$ such that $\widehat{c}=c^{\widehat{\alpha}}$, only if he knows a set $\{a_i\}_{i\in\{0\}\cup\Lambda}$ such that $c=\prod_{i\in\{0\}\cup\Lambda}\left(g_t^{x^i}\right)^{a_i}$. Note that t defines the generators $g_1\in\mathbb{G}_1\setminus\{1\}$ and $g_2\in\mathbb{G}_2\setminus\{1\}$. The Λ-PSDL and Λ-PKE assumptions from [Lip12a] are formally explained in Sections 2.2.2 and 2.2.5. The lifted BBS cryptosystem requires the decision linear (DLIN) assumption. This assumption states that for either a bilinear group $\mathbb{G}_1$ for $t=1$ or $\mathbb{G}_2$ for $t=2$, the adversary is unable to distinguish between $g_t^{\sigma+\tau}$ and g_t^z for random $\sigma,\ \tau,\ z$, when the adversary input is $\left(f,h,f^{\sigma},h^{\tau}\right)$ with $f,\ h$ taken randomly from the bilinear group corresponding to t. Therefore, the Λ-PSDL, the Λ-PKE, and the DLIN assumptions are needed for the subargument of this section.

System parameters: $\Lambda=\{\lambda_i\}_{i\in\mathbb{Z}_n}\subset\mathbb{N}$ such that $\forall i<j\colon 0<\lambda_i<\lambda_j$.

Common reference string generator $\mathrm{Gen}_{\mathsf{crs}\_sub}(1^\kappa)$**:**

Set $\mathsf{param_{bp}}:=(p,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_{\mathrm{T}},e)\leftarrow\mathsf{PG}_a(1^\kappa)$;
Generate random $\alpha_g,\alpha_f,\alpha_h,\widehat{\alpha},\alpha_{g/c},x\in_R\mathbb{Z}_p^*$;
Let $g_1\in_R\mathbb{G}_1\setminus\{1\}$ and $g_2\in_R\mathbb{G}_2\setminus\{1\}$,

denote
$$g_{1,\lambda_0}\leftarrow g_1^{x^{\lambda_0}},\quad \mathring{g}_1\leftarrow g_1^{\alpha_g},\quad \widehat{g}_1\leftarrow g_1^{\widehat{\alpha}},\quad \widehat{g}_{1,\lambda_0}\leftarrow g_{1,\lambda_0}^{\widehat{\alpha}},$$
$$g_{2,\lambda_0}\leftarrow g_2^{x^{\lambda_0}},\quad \mathring{g}_2\leftarrow g_2^{\alpha_g},\quad \widehat{g}_2\leftarrow g_2^{\widehat{\alpha}},\quad \widehat{g}_{2,\lambda_0}\leftarrow g_{2,\lambda_0}^{\widehat{\alpha}},$$
$$\mathring{g}_{1,f}\leftarrow g_1^{\alpha_f},\quad \mathring{g}_{1,h}\leftarrow g_1^{\alpha_h},\quad \mathring{g}_{1,g/c}\leftarrow g_1^{\alpha_{g/c}},$$
$$\mathring{g}_{2,f}\leftarrow g_2^{\alpha_f},\quad \mathring{g}_{2,h}\leftarrow g_2^{\alpha_h},\quad \mathring{g}_{2,g/c}\leftarrow g_2^{\alpha_{g/c}},$$

and $\mathring{g}_{1,\lambda_0}\leftarrow g_{1,\lambda_0}^{\alpha_{g/c}}$;

Set the common reference string

$$\mathsf{crs}\leftarrow\left\{\begin{matrix}\mathsf{param_{bp}};\ g_1,\ g_{1,\lambda_0},\ \mathring{g}_1,\ \widehat{g}_1,\ \widehat{g}_{1,\lambda_0},\ \mathring{g}_{1,f},\ \mathring{g}_{1,h},\ \mathring{g}_{1,g/c},\ \mathring{g}_{1,\lambda_0},\\ g_2,\ g_{2,\lambda_0},\ \mathring{g}_2,\ \widehat{g}_2,\ \widehat{g}_{2,\lambda_0},\ \mathring{g}_{2,f},\ \mathring{g}_{2,h},\ \mathring{g}_{2,g/c}\end{matrix}\right\};$$

A third party also creates $\mathsf{sk}:=(\mathsf{sk}_1,\mathsf{sk}_2)\in_R(\mathbb{Z}_p^*)^2$,
and sets $\mathsf{pk}:=(f,h,\mathring{f},\mathring{h})\leftarrow(g_1^{1/\mathsf{sk}_1},g_1^{1/\mathsf{sk}_2},\mathring{g}_{1,f}^{1/\mathsf{sk}_1},\mathring{g}_{1,h}^{1/\mathsf{sk}_2})$.

Common inputs: $(\mathsf{crs};\ \mathsf{pk},A_g,A_f,A_h,A_c)$,
where $(A_g,A_f,A_h)=\left(g_1^{r_f+r_h+\sigma},\ f^{r_f},\ h^{r_h}\right)$,
and $A_c=g_1^{\tilde{r}}g_{1,\lambda_0}^{\sigma}=g_1^{r_f+r_h+r_c}g_{1,\lambda_0}^{\sigma}$.

Protocol 5.1a – Setup of the equality subargument
of a knowledge committed value and its lifted BBS encryption

Protocol explanation. The subargument of this section, described in Protocol 5.1a and 5.1b, is constructed by combining ideas from [GS08, GS12a] and [Gro10, Lip12a]. Intuitively, for

Argument generated by the prover:

$$\text{NIZK-PK}\left\{(\sigma, r_f, r_h, \tilde{r}) : \left(A_g, A_f, A_h\right) = \left(g_1^{r_f+r_h+\sigma},\ f^{r_f},\ h^{r_h}\right) \ \wedge \ A_c = g_1^{\tilde{r}} g_{1,\lambda_0}^{\sigma}\right\}$$

Set $r_c \leftarrow (\tilde{r} - r_f - r_h)$, $\widehat{A}_c \leftarrow \widehat{g}_1^{\tilde{r}} \widehat{g}_{1,\lambda_0}^{\sigma}$, $\mathring{A}_{g/c} \leftarrow \mathring{g}_{1,g/c}^{\sigma - r_c} \mathring{g}_{1,\lambda_0}^{-\sigma}$,
$\left(\mathring{A}_g, \mathring{A}_f, \mathring{A}_h\right) \leftarrow \left(\mathring{g}_1^{r_f+r_h+\sigma}, \mathring{f}^{r_f}, \mathring{h}^{r_h}\right)$.
Pick random $R_f, R_h \in_R \mathbb{Z}_p^*$.
Set $\left(C_f,\ \widehat{C}_f\right) \leftarrow \left(g_2^{R_f} g_{2,\lambda_0}^{r_f},\quad \widehat{g}_2^{R_f} \widehat{g}_{2,\lambda_0}^{r_f}\right) \in \mathbb{G}_2^2$,
$\left(C_h,\ \widehat{C}_h\right) \leftarrow \left(g_2^{R_h} g_{2,\lambda_0}^{r_h},\quad \widehat{g}_2^{R_h} \widehat{g}_{2,\lambda_0}^{r_h}\right) \in \mathbb{G}_2^2$,
$\left(\pi_g,\ \mathring{\pi}_g\right) \leftarrow \left(g_1^{\tilde{r}+R_f+R_h},\quad \mathring{g}_1^{\tilde{r}+R_f+R_h}\right) \in \mathbb{G}_1^2$,
$\left(\pi_f,\ \mathring{\pi}_f\right) \leftarrow \left(f^{R_f},\ \mathring{f}^{R_f}\right) \in \mathbb{G}_1^2$,
$\left(\pi_h,\ \mathring{\pi}_h\right) \leftarrow \left(h^{R_h},\ \mathring{h}^{R_h}\right) \in \mathbb{G}_1^2$.
Send to the verifier the argument:

$$\pi^{ce} \leftarrow \left(\mathring{A}_g, \mathring{A}_f, \mathring{A}_h,\ \widehat{A}_c,\ \pi_g, \mathring{\pi}_g,\ C_f, \widehat{C}_f, \pi_f, \mathring{\pi}_f,\ C_h, \widehat{C}_h, \pi_h, \mathring{\pi}_h,\ \mathring{A}_{g/c}\right).$$

Verification $\left(\mathsf{crs}; (A_g, A_f, A_h, A_c), \pi^{ce}\right)$**:**

Verify that

$e(\mathring{f}, g_2) \stackrel{?}{=} e(f, \mathring{g}_{2,f})$, $\quad e(\mathring{h}, g_2) \stackrel{?}{=} e(h, \mathring{g}_{2,h})$,
$e(\mathring{A}_g, g_2) \stackrel{?}{=} e(A_g, \mathring{g}_2)$, $\quad e(\mathring{A}_f, g_2) \stackrel{?}{=} e(A_f, \mathring{g}_{2,f})$,
$e(\mathring{A}_h, g_2) \stackrel{?}{=} e(A_h, \mathring{g}_{2,h})$, $\quad e(\widehat{A}_c, g_2) \stackrel{?}{=} e(A_c, \widehat{g}_2)$,
$e(\mathring{\pi}_g, g_2) \stackrel{?}{=} e(\pi_g, \mathring{g}_2)$, $\quad e(\mathring{\pi}_f, g_2) \stackrel{?}{=} e(\pi_f, \mathring{g}_{2,f})$, $\quad e(\mathring{\pi}_h, g_2) \stackrel{?}{=} e(\pi_h, \mathring{g}_{2,h})$,
$e(g_1, \widehat{C}_f) \stackrel{?}{=} e(\widehat{g}_1, C_f)$, $\quad e(g_1, \widehat{C}_h) \stackrel{?}{=} e(\widehat{g}_1, C_h)$,
and $e(\mathring{A}_{g/c}, g_2) \stackrel{?}{=} e(A_g / A_c, \mathring{g}_{2,g/c})$.

Verify that
$e(f, C_f) \stackrel{?}{=} e(\pi_f, g_2) \cdot e(A_f, g_{2,\lambda_0})$,
$e(h, C_h) \stackrel{?}{=} e(\pi_h, g_2) \cdot e(A_h, g_{2,\lambda_0})$,
and $e(g_1, C_f C_h) \stackrel{?}{=} e(\pi_g A_c^{-1}, g_2) \cdot e(A_g, g_{2,\lambda_0})$.

Protocol 5.1b – Equality subargument of a knowledge committed value and its lifted BBS encryption

every multi-exponentiation $h_1^{\sigma_1} \ldots h_m^{\sigma_m} = t$ that needs to be proven, a verification equation $e(h_1, \mathrm{Com}(\sigma_1)) \cdot \cdots \cdot e(h_m, \mathrm{Com}(\sigma_m)) = e(\pi, g_2) e(t, \mathrm{Com}(1))$ is provided, where π "compensates" for the fact that $\mathrm{Com}(\sigma_i)$ are probabilistic commitments. In addition, knowledge commitments are used (though for small values 0 or 1 of n) so that all committed values can be extracted. Since the argument uses three committed values (σ, r_f, and r_h) and three equations, according to Figure 4 of [GS07][2] (the full version of [GS08, GS12a]), the corresponding pure Groth-Sahai argument has a length of 15 group elements. The subargument presented here has the same length, but is computationally more efficient.

Theorem 5.1

The argument in Protocol 5.1b is a perfectly complete and perfectly zero-knowledge argument that for some $\sigma \in \mathbb{Z}_p$, $\tilde{r}, r_f, r_h \in \mathbb{Z}_p^*$, $A_c = g_1^{\tilde{r}} g_{1,\lambda_0}^{\sigma}$ and $(A_g, A_f, A_h) = (g^{r_f + r_h + \sigma}, f^{r_f}, h^{r_h})$. If the $\{\lambda_0\}$-PSDL assumption and the $\{\lambda_0\}$-PKE assumption (in both $\mathbb{G}_1$ and $\mathbb{G}_2$) hold, then this argument is computationally sound.

Clearly, this argument has a CRS of length $\Theta(1)$, its argument consists of 11 elements of $\mathbb{G}_1$ and 4 elements of $\mathbb{G}_2$. The computational complexity of the prover is dominated by 13 exponentiations in $\mathbb{G}_1$ and 8 exponentiations in $\mathbb{G}_2$. The computational complexity of the verifier is dominated by 33 pairings.

Proof

To show that the argument described in Protocol 5.1b is a NIZK argument, three security properties are proven: *perfect completeness*, *computational soundness*, and *perfect zero-knowledge*.

PERFECT COMPLETENESS: all verification equations hold as follows:

$$
\begin{array}{lllll}
e(\mathring{f}, g_2) = e(f, g_2)^{\alpha_f} = e(f, \mathring{g}_{2,f}); & & e(\mathring{h}, g_2) = e(h, g_2)^{\alpha_h} = e(h, \mathring{g}_{2,h}); \\
e(\mathring{A}_g, g_2) = e(A_g, g_2)^{\alpha_g} = e(A_g, \mathring{g}_2); & & e(\mathring{A}_f, g_2) = e(A_f, g_2)^{\alpha_f} = e(A_f, \mathring{g}_{2,f}); \\
e(\mathring{A}_h, g_2) = e(A_h, g_2)^{\alpha_h} = e(A_h, \mathring{g}_{2,h}); & & e(\widehat{A}_c, g_2) = e(A_c, g_2)^{\widehat{\alpha}} = e(A_c, \widehat{g}_2); \\
e(\mathring{\pi}_g, g_2) = e(\pi_g, g_2)^{\alpha_g} = e(\pi_g, \mathring{g}_2); & & e(\mathring{\pi}_f, g_2) = e(\pi_f, g_2)^{\alpha_f} = e(\pi_f, \mathring{g}_{2,f}); \\
e(\mathring{\pi}_h, g_2) = e(\pi_h, g_2)^{\alpha_h} = e(\pi_h, \mathring{g}_{2,h}); & & \\
e(g_1, \widehat{C}_f) = e(g_1, C_f)^{\widehat{\alpha}} = e(\widehat{g}_1, C_f); & & e(g_1, \widehat{C}_h) = e(g_1, C_h)^{\widehat{\alpha}} = e(\widehat{g}_1, C_h);
\end{array}
$$

$$
\begin{array}{lll}
e(\mathring{A}_{g/c}, g_2) & = e(\mathring{g}_{1,g/c}^{\sigma - r_c} \mathring{g}_{1,\lambda_0}^{-\sigma}, g_2) & = e(g_1^{(\sigma - r_c)\cdot\alpha_{g/c} - \sigma\cdot\alpha_{g/c}\cdot x^{\lambda_0}}, g_2) \quad = e(g_1^{\sigma - r_c} g_1^{-\sigma x^{\lambda_0}}, g_2)^{\alpha_{g/c}} \\
& = e(g_1^{\sigma - r_c} g_{1,\lambda_0}^{-\sigma}, \mathring{g}_{2,g/c}) & = e(A_g / A_c, \mathring{g}_{2,g/c}).
\end{array}
$$

[2] This publication has several versions. The one referred here is the version published in April 2016.

$$\begin{aligned}
e(f, C_f) = e(f, g_2^{R_f} g_{2,\lambda_0}^{r_f}) &= e(f, g_2^{R_f}) \cdot e(f, g_{2,\lambda_0}^{r_f}) \\
&= e(f^{R_f}, g_2) \cdot e(f^{r_f}, g_{2,\lambda_0}) \\
&= e(\pi_f, g_2) \cdot e(A_f, g_{2,\lambda_0}). \\
e(h, C_h) = e(h, g_2^{R_h} g_{2,\lambda_0}^{r_h}) &= e(h, g_2^{R_h}) \cdot e(h, g_{2,\lambda_0}^{r_h}) \\
&= e(h^{R_h}, g_2) \cdot e(h^{r_h}, g_{2,\lambda_0}) \\
&= e(\pi_h, g_2) \cdot e(A_h, g_{2,\lambda_0}).
\end{aligned}$$

$$\begin{aligned}
e(A_c \pi_g^{-1}, g_2) \cdot e(g_1, C_f C_h) &= e(g_1^{\widetilde{r}} g_{1,\lambda_0}^{\sigma} \cdot g_1^{-\widetilde{r}-R_f-R_h}, g_2) \cdot e(g_1, g_2^{R_f+R_h}) \cdot e(g_1, g_{2,\lambda_0}^{r_f+r_h}) \\
&= e(g_{1,\lambda_0}^{\sigma} \cdot g_1^{-R_f-R_h}, g_2) \cdot e(g_1^{R_f+R_h}, g_2) \cdot e(g_1^{r_f+r_h}, g_{2,\lambda_0}) \\
&= e(g_1^{\sigma}, g_{2,\lambda_0}) \cdot e(g_1^{r_f+r_h}, g_{2,\lambda_0}) \\
&= e(g_1^{r_f+r_h+\sigma}, g_{2,\lambda_0}) \\
&= e(A_g, g_{2,\lambda_0}).
\end{aligned}$$

COMPUTATIONAL SOUNDNESS: By the $\{\lambda_0\}$-PKE assumption in $\mathbb{G}_1/\mathbb{G}_2$, one can open the next values:

$$\begin{aligned}
(A_c, \widehat{A}_c) &= \left(g_1^{\widetilde{r}} g_{1,\lambda_0}^{\sigma}, \widehat{g}_1^{\widetilde{r}} \widehat{g}_{1,\lambda_0}^{\sigma}\right), & (A_g / A_c, \mathring{A}_{g/c}) &= \left(g_1^{\sigma'_r} g_{1,\lambda_0}^{-\sigma'}, \mathring{g}_{1,g/c}^{\sigma'_r} \mathring{g}_{1,\lambda_0}^{-\sigma'}\right), \\
(A_g, \mathring{A}_g) &= \left(g_1^{\sigma''}, \mathring{g}_1^{\sigma''}\right), & (\pi_g, \mathring{\pi}_g) &= \left(g_1^{r''_\sigma}, \mathring{g}_1^{r''_\sigma}\right), \\
(A_f, \mathring{A}_f) &= \left(f^{r_f}, \mathring{f}^{r_f}\right), & (\pi_f, \mathring{\pi}_f) &= \left(g_1^{r''_f}, \mathring{g}_{1,f}^{r''_f}\right), \\
(A_h, \mathring{A}_h) &= \left(h^{r_h}, \mathring{h}^{r_h}\right), & (\pi_h, \mathring{\pi}_h) &= \left(g_1^{r''_h}, \mathring{g}_{1,h}^{r''_h}\right), \\
(C_f, \widehat{C}_f) &= \left(g_2^{R_f} g_{2,\lambda_0}^{r'_f}, \widehat{g}_2^{R_f} \widehat{g}_{2,\lambda_0}^{r'_f}\right), & \text{and } (C_h, \widehat{C}_h) &= \left(g_2^{R_h} g_{2,\lambda_0}^{r'_h}, \widehat{g}_2^{R_h} \widehat{g}_{2,\lambda_0}^{r'_h}\right).
\end{aligned}$$

Since $A_c = g_1^{\widetilde{r}} g_{1,\lambda_0}^{\sigma}$, $A_g = g_1^{\sigma''}$, and $A_g / A_c = g_1^{\sigma'_r} g_{1,\lambda_0}^{-\sigma'}$, it implies that $g_1^{\sigma''} = g_1^{\sigma'_r + \widetilde{r}} g_{1,\lambda_0}^{\sigma-\sigma'}$. Thus, if $\sigma \neq \sigma'$, an adversary can compute $x^{\lambda_0} \leftarrow (\sigma'' - \sigma'_r - \widetilde{r})/(\sigma - \sigma')$, and from this compute x and thus break the $\{\lambda_0\}$-PSDL assumption. (To verify whether x is the correct root, the adversary can check that $g_1^{x^{\lambda_0}} = g_{1,\lambda_0}$.) Thus $\sigma = \sigma'$, and thus also $\sigma'' = \sigma'_r + \widetilde{r}$ and $A_g = g_1^{\sigma'_r + \widetilde{r}}$.

As $C_f = g_2^{R_f} g_{2,\lambda_0}^{r'_f}$, $\pi_f = g_1^{r''_f}$, $A_f = f^{r_f}$, and $e(f, C_f) = e(\pi_f, g_2) \cdot e(A_f, g_{2,\lambda_0})$, this implies that $e(f, g_2^{R_f} g_{2,\lambda_0}^{r'_f}) = e(g_1^{r''_f}, g_2) e(f^{r_f}, g_2^{x^{\lambda_0}})$, for an unknown x. Taking the discrete logarithm in both sides of the last equation, the following is obtained:

$$\begin{aligned}
R_f/\mathsf{sk}_1 + r'_f x^{\lambda_0}/\mathsf{sk}_1 &= r''_f + r_f x^{\lambda_0}/\mathsf{sk}_1 \\
\iff (r_f - r'_f) x^{\lambda_0} &= R_f - r''_f \cdot \mathsf{sk}_1.
\end{aligned}$$

If $r_f \neq r'_f$, then x^{λ_0} could be computed, and from it x can be derived which would break the

$\{\lambda_0\}$-PSDL assumption. Therefore, $r_f = r'_f$ and also $C_f = g_2^{R_f} g_{2,\lambda_0}^{r_f}$. Moreover, $\pi_f = g_1^{r''_f} = f^{R_f}$. With the same reasoning, $r_h = r'_h$ is obtained and therefore $C_h = g_1^{R_h} g_{1,\lambda_0}^{r_h}$ and $\pi_h = h^{R_h}$.

As $C_f = g_2^{R_f} g_{2,\lambda_0}^{r_f}$, $C_h = g_2^{R_h} g_{2,\lambda_0}^{r_h}$, $\pi_g = g_1^{r''_\sigma}$, $A_c = g_1^{\tilde{r}} g_{1,\lambda_0}^{\sigma}$, $A_g = g_1^{\sigma'_r + \tilde{r}}$, and $e(g_1, C_f C_h) = e(\pi_g A_c^{-1}, g_2) \cdot e(A_g, g_{2,\lambda_0})$, this implies that

$$e(g_1, g_2^{R_f+R_h+(r_f+r_h)x^{\lambda_0}}) = e(g_1^{r''_\sigma} g_1^{-\tilde{r}} g_{1,\lambda_0}^{-\sigma}, g_2) \cdot e(g_1^{\sigma'_r+\tilde{r}}, g_{2,\lambda_0}) = e(g_1^{r''_\sigma-\tilde{r}+(\sigma'_r-\sigma+\tilde{r})x^{\lambda_0}}, g_2)$$

for an unknown x. Taking the discrete logarithm in both sides of the last equation, the equation becomes $R_f+R_h+(r_f+r_h)x^{\lambda_0} = r''_\sigma - \tilde{r} + (\sigma'_r - \sigma + \tilde{r})x^{\lambda_0}$. Again, if $(r_f+r_h) \neq (\sigma'_r - \sigma + \tilde{r})$, then it is possible to compute x^{λ_0} and thus also x. Thus, $\sigma'_r + \tilde{r} = r_f + r_h + \sigma$, and thus also $r''_\sigma = \tilde{r} + R_f + R_h$. This means that $A_c = g_1^{\tilde{r}} g_{1,\lambda_0}^{\sigma}$ and $(A_g, A_f, A_h) = (g_1^{r_f+r_h+\sigma}, f^{r_f}, h^{r_h})$.

PERFECT ZERO-KNOWLEDGE: to prove computational zero-knowledge, the following simulator (Sim_1, Sim_2) is constructed. Sim_1 creates a CRS according to the protocol together with a trapdoor $\mathsf{td} = (\alpha_g, \alpha_f, \alpha_h, \widehat{\alpha}, \alpha_{g/c}, x)$. On input $(\mathsf{crs}, \mathsf{pk}, A_g, A_f, A_h, A_c, \mathsf{td})$, Sim_2 picks $z_f, z_h \in_R \mathbb{Z}_p$, then sets $C_f \leftarrow g_2^{z_f}$, $\pi_f \leftarrow f^{z_f} A_f^{-x^{\lambda_0}}$, $C_h \leftarrow g_2^{z_h}$, $\pi_h \leftarrow h^{z_h} A_h^{-x^{\lambda_0}}$, and $\pi_g \leftarrow g_1^{z_f+z_h} A_c A_g^{-x^{\lambda_0}}$. The elements C_f and C_h have the same distribution as the honestly generated ones, as z_f and z_h have respectively the same distributions as $\left(R_f + r_f x^{\lambda_0}\right)$ and $\left(R_h + r_h x^{\lambda_0}\right)$. The success of the verification equations can be checked for the choices of π_f, π_h, and π_g. For example:

$$\begin{aligned} e(\pi_f, g_2) e(A_f, g_{2,\lambda_0}) =& e(f^{z_f} A_f^{-x^{\lambda_0}}, g_2) \cdot e(A_f, g_{2,\lambda_0}) = e(f^{z_f}, g_2) e(A_f^{-x^{\lambda_0}}, g_2) e(A_f, g_{2,\lambda_0}) \\ =& e(f^{z_f}, g_2) = e(f, C_f), \end{aligned}$$

and finally, $e(A_c \pi_g^{-1}, g_2) \cdot e(g_1, C_f C_h) = e(g_1^{-z_f-z_h} A_g^{x^{\lambda_0}}, g_2) \cdot e(g_1, g_2^{z_f+z_h}) = e(A_g, g_{2,\lambda_0})$. Hence π_f, π_h, and π_g will be accepted by the verification. Moreover, because these verification equations fix π_f, π_h, and π_g uniquely (given the inputs of Sim_2 and as C_f, C_h are set with the correct distribution), the tuple $\left(C_f, \pi_f, C_h, \pi_h, \pi_g\right)$ comes from the correct distribution

Sim_2 creates the knowledge elements $(\mathring{A}_g, \mathring{A}_f, \mathring{A}_h, \widehat{A}_c, \mathring{\pi}_g, \widehat{C}_f, \mathring{\pi}_f, \widehat{C}_h, \mathring{\pi}_h, \mathring{A}_{g/c})$ by using the trapdoor, which will result in elements with the same distribution as the honestly generated ones. For example, $\mathring{A}_{g/c} \leftarrow (A_g / A_c)^{\alpha_{g/c}}$ and $\mathring{A}_g \leftarrow A_g^{\alpha_g}$. The simulated argument will thus be:

$$\pi^{ce} \leftarrow \left(\mathring{A}_g, \mathring{A}_f, \mathring{A}_h,\ \widehat{A}_c,\ \pi_g, \mathring{\pi}_g,\ C_f, \widehat{C}_f, \pi_f, \mathring{\pi}_f,\ C_h, \widehat{C}_h, \pi_h, \mathring{\pi}_h,\ \mathring{A}_{g/c}\right).$$

As the simulator Sim_2 outputs an accepting argument π^{ce} with the exact same distribution as the one from the honest prover, the argument achieves perfect zero-knowledge. ■

Remark: Professor Groth pointed out that the initial description [CLZ12] of the equality subargument had the value $\mathring{A}_{g/c}$ set to $\mathring{A}_{g/c} = \mathring{g}_{1,g/c}^{\sigma}$. This is obviously an issue for the zero-knowledge property, as a simulator Sim_2, without the knowledge of σ, cannot produce the element $\mathring{A}_{g/c}$ with the same distribution as the prover would. Furthermore, perfect zero-

knowledge is achieved with the corrected version present in this thesis, contrarily to the claimed computational zero-knowledge from the initial description.

5.5 Lifted BBS Encryption Based Non-Interactive Range Proof

In the NIZK range proof presented in this section, the prover has an encrypted $\sigma \in \mathbb{Z}_p$, and his aim is to convince the verifier that $\sigma \in [0, H]$. The encryption protocol used comes from the lifted BBS cryptosystem $(\text{Gen}_{pkc}, \text{Enc}, \text{Dec})$, as described in Section 2.4.4. It can be thought of as a perfectly binding commitment scheme as long as decryption is not necessary. The Lipmaa computationally binding knowledge commitment scheme $(\text{Gen}_{com}, \text{Com}, \text{Open})$, as described in Section 2.4.2, will be used here to obtain a sublinear argument. Furthermore, the range proof is based on results from Section 4.4. As a reminder, for $H > 0$, $u > 1$, and $\ell(u, H)$ defined as in Section 4.4,Theorem 4.10 states that $\sigma \in [0, H]$ if and only if for some $\sigma_i \in [0, u-1]$, the following holds:

$$(u-1)\sigma = \sum_{i=0}^{\ell(u,(u-1)H)-1} G_i \sigma_i,$$

where $G_i \in \mathbb{Z}$ are values defined by equations 4.6 and 4.7. Moreover, for the binary decomposition case (where $u = 2$), Lemma 4.8 implies that $\ell \leqslant \log_2 H$, which in turn implies that $\ell \leqslant \lfloor \log_2 H \rfloor$ as ℓ must be an integer. Therefore, from Corollary 4.12 and Theorem 4.10, it can be concluded that $\sigma \in [0, H]$ if and only if for some $\sigma_i \in [0, 1]$, the following holds:

$$\sigma = \sum_{i=0}^{\lfloor \log_2 H \rfloor - 1} \left\lfloor \frac{H + 2^i}{2^{i+1}} \right\rfloor \sigma_i.$$

The precise values of $\ell(u, H)$ and G_i are not important for the protocol explanation. It is sufficient to know that they can be efficiently evaluated.

Computational assumptions. The NIZK range proof of this section uses the Hadamard product argument (see Section 2.4.7), the Lipmaa permutation argument (see Section 2.4.8), and the subargument for lifted BBS encryption of a Lipmaa knowledge committed value (explained in the previous section). The following assumptions are therefore required for the NIZK range proof presented in this section: the bilinear groups associated computational hardness assumptions (see Section 2.1.3), the Λ-power symmetric discrete logarithm (Λ-PSDL) assumption, the Λ-power knowledge of exponent (Λ-PKE) assumption, and the decision linear (DLIN) assumption. The Λ-PSDL and the DLIN assumptions are explained in Section 2.2.2. The Λ-PKE assumption is explained in Section 2.2.5.

System parameters: $H, u, G_i, \ell, \ell_v := \lfloor \log_2(u-1) \rfloor$, and $G'_j := \lfloor (u-1+2^j)/2^{j+1} \rfloor$,
a progression-free set $\Lambda = \{\lambda_i\}_{i \in \mathbb{Z}_\ell} \subset \mathbb{N}$ such that $\forall i < j\colon 0 < \lambda_i < \lambda_j$,
$\widehat{\Lambda} := \{0\} \cup \Lambda \cup 2\hat{}\Lambda$,
$\widetilde{\Lambda} := \Lambda \cup \{2\lambda_k - \lambda_j\}_{i,k \in \{0,\ldots,\ell-1\}} \cup 2\hat{}\Lambda \cup \left(\{2\lambda_k + \lambda_i - \lambda_j\}_{i,j,k \in \{0,\ldots,\ell-1\} \wedge i \neq j} \setminus 2 \cdot \Lambda\right)$.

Common reference string generator $\mathrm{Gen}_{\mathsf{crs\_}nirp}(1^\kappa)$**:**
Set $\mathsf{param}_{\mathsf{bp}} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathsf{PG}_a(1^\kappa)$;
Generate random $\widehat{\alpha}, \widetilde{\alpha}, \alpha_g, \alpha_f, \alpha_h, \alpha_{g/c}, x \in_R \mathbb{Z}_p^*$;
Let $g_1 \in_R \mathbb{G}_1 \setminus \{1\}$ and $g_2 \in_R \mathbb{G}_2 \setminus \{1\}$,

denote $g_{t,s} \leftarrow g_t^{x^s}$, $\widehat{g}_{t,s} \leftarrow g_t^{\widehat{\alpha} x^s}$, $\widetilde{g}_{t,s} \leftarrow g_t^{\widetilde{\alpha} x^s}$, where $s \in \{0\} \cup \widetilde{\Lambda}$, and $t \in \{1,2\}$,

$$\mathring{g}_1 \leftarrow g_1^{\alpha_g}, \quad \mathring{g}_{1,g/c} \leftarrow g_1^{\alpha_{g/c}}, \quad \mathring{g}_{1,f} \leftarrow g_1^{\alpha_f}, \quad \mathring{g}_{1,h} \leftarrow g_1^{\alpha_h},$$
$$\mathring{g}_2 \leftarrow g_2^{\alpha_g}, \quad \mathring{g}_{2,g/c} \leftarrow g_2^{\alpha_{g/c}}, \quad \mathring{g}_{2,f} \leftarrow g_2^{\alpha_f}, \quad \mathring{g}_{2,h} \leftarrow g_2^{\alpha_h},$$

and $\mathring{g}_{1,\lambda_0} \leftarrow g_{1,\lambda_0}^{\alpha_{g/c}}$;

Set $D \leftarrow \prod_{i=0}^{\ell-1} g_{2,\lambda_i}$, $\widetilde{D} \leftarrow D^{\widetilde{\alpha}}$, $E_{\mathsf{rot}} \leftarrow \prod_{i=0}^{\ell-1} g_{2,2\lambda_{\mathsf{rot}(i)}-\lambda_i}$, $\widetilde{E}_{\mathsf{rot}} \leftarrow E_{\mathsf{rot}}^{\widetilde{\alpha}}$
and $\left(T^*, \widehat{T}^*, T_2^*\right) \leftarrow \left(\prod_{i=0}^{\ell-1} g_{1,\lambda_i}^{T_{\Lambda,\mathsf{rot}}(i)}, \prod_{i=0}^{\ell-1} \widehat{g}_{1,\lambda_i}^{T_{\Lambda,\mathsf{rot}}(i)}, \prod_{i=0}^{\ell-1} g_{2,\lambda_i}^{T_{\Lambda,\mathsf{rot}}(i)}\right)$,
where $T_{\Lambda,\mathsf{rot}}(i) = \left|\left\{j \in \mathbb{Z}_\ell : 2\lambda_i + \lambda_j = 2\lambda_{\mathsf{rot}(j)} + \lambda_{\mathsf{rot}^{-1}(i)}\right\}\right|$;

Set the common reference string

$$\mathsf{crs} \leftarrow \left\{ \begin{array}{l} \mathsf{param}_{\mathsf{bp}};\ \left(g_{1,s}, \widehat{g}_{1,s}, \widetilde{g}_{1,s}\right)_{s \in \{0\} \cup \Lambda},\ g_2,\ \left(\widehat{g}_{2,s}\right)_{s \in \widehat{\Lambda}}, \\ \qquad \left(g_{2,s}, \widetilde{g}_{2,s}\right)_{s \in \widetilde{\Lambda}},\ \left\{\mathring{g}_t, \mathring{g}_{t,g/c}, \mathring{g}_{t,f}, \mathring{g}_{t,h}\right\}_{t \in \{1,2\}}, \\ \qquad \mathring{g}_{1,\lambda_0},\ D,\ \widetilde{D},\ E_{\mathsf{rot}},\ \widetilde{E}_{\mathsf{rot}},\ T^*,\ \widehat{T}^*,\ T_2^* \end{array} \right\};$$

Set $\mathsf{crs}_1 \leftarrow \left\{\mathsf{param}_{\mathsf{bp}};\ \left(g_{1s}, \widehat{g}_{1s}, \widetilde{g}_{1s}\right)_{s \in \{0\} \cup \Lambda}\right\} \subset \mathsf{crs}$,
$\widehat{\mathsf{crs}}_1 \leftarrow \left\{\mathsf{param}_{\mathsf{bp}};\ \left(g_{1s}, \widehat{g}_{1s}\right)_{s \in \{0\} \cup \Lambda}\right\} \subset \mathsf{crs}$, and
$\widetilde{\mathsf{crs}}_1 \leftarrow \left\{\mathsf{param}_{\mathsf{bp}};\ \left(g_{1s}, \widetilde{g}_{1s}\right)_{s \in \{0\} \cup \Lambda}\right\} \subset \mathsf{crs}$;
The prover creates a secret key $\mathsf{sk} := (\mathsf{sk}_1, \mathsf{sk}_2) \in_R (\mathbb{Z}_p^*)^2$,
and sets $\mathsf{pk} := (f, h, \mathring{f}, \mathring{h}) \leftarrow \left(g_1^{1/\mathsf{sk}_1},\ g_1^{1/\mathsf{sk}_2},\ \mathring{g}_{1,f}^{1/\mathsf{sk}_1},\ \mathring{g}_{1,h}^{1/\mathsf{sk}_2}\right)$.

Common inputs: $(\mathsf{pk}, A_g, A_f, A_h, A_c, \widehat{A}_c)$, where $(A_g, A_f, A_h) = (g_1^{r+\sigma}, f^{r_f}, h^{r_h})$
and $(A_c, \widehat{A}_c) = \left(g_1^{\widetilde{r}} g_{1,\lambda_1}^{\sigma}, \widehat{g}_1^{\widetilde{r}} \widehat{g}_{1,\lambda_1}^{\sigma}\right)$, for $r = r_f + r_h$ and $\widetilde{r} = r + r_c$.

Protocol 5.2a – Setup of the non-interactive range proof protocol for the range $[0, H]$

Argument generated by the prover:

$$\text{NIZK-PK}\left\{(\sigma, r_f, r_h) : \left(A_g, A_f, A_h\right) = \left(g_1^{r_f+r_h+\sigma},\ f^{r_f},\ h^{r_h}\right) \ \wedge\ \sigma \in [0, H]\right\}$$

1. Compute $(\sigma_0, \ldots, \sigma_{\ell-1}) \in \mathbb{Z}_u^\ell$ such that $(u-1)\,\sigma = \sum_{i=0}^{\ell-1} G_i \sigma_i$.
2. For $i \in \mathbb{Z}_\ell$ compute $\left(\sigma'_{0,i}, \ldots, \sigma'_{\ell_v-1,i}\right) \in \mathbb{Z}_2^{\ell_v}$ such that $\sigma_i = \sum_{j=0}^{\ell_v-1} G'_j \cdot \sigma'_{j,i}$.
3. For $j \in \mathbb{Z}_{\ell_v}$:
   - Let $r_j \in_R \mathbb{Z}_p^*$, $\left(B'_j, \widehat{B}'_j\right) \leftarrow \text{Com}^1\left(\widehat{\text{crs}}_1; \sigma'_{j,0}, \ldots, \sigma'_{j,\ell-1}; r_j\right)$, $B'_{j,2} \leftarrow g_2^{r_j} \cdot \prod_{i=0}^{\ell-1} g_{2,\lambda_i}^{\sigma'_{j,i}}$.
   - Create a Hadamard product argument $\left(\pi'_j, \widehat{\pi}'_j\right)$ for $\left[\!\left[\left(B'_j, \widehat{B}'_j\right)\right]\!\right] = \left[\!\left[\left(B'_j, \widehat{B}'_j\right)\right]\!\right] \circ \left[\!\left[\left(B'_j, \widehat{B}'_j, B'_{j,2}\right)\right]\!\right]$.
4. For $i \in \mathbb{Z}_\ell$, let $c_i \leftarrow \sum_{k=i}^{\ell-1} G_k \sigma_k$.
5. Set $r'_0, r'_1, r'_2 \in_R \mathbb{Z}_p^*$, $\left(B^\dagger, \widehat{B}^\dagger\right) \leftarrow \text{Com}^1\left(\widehat{\text{crs}}_1; G_0\sigma_0, \ldots, G_{\ell-1}\sigma_{\ell-1}; r'_0\right)$, $\left(C, \widehat{C}, \widetilde{C}\right) \leftarrow \text{Com}^1\left(\text{crs}_1; \boldsymbol{c}; r'_1\right)$, and $\left(C_{\text{rot}}, \widehat{C}_{\text{rot}}, \widetilde{C}_{\text{rot}}\right) \leftarrow \text{Com}^1\left(\text{crs}_1; c_1, \ldots, c_{\ell-2}, c_{\ell-1}, c_0; r'_2\right)$.
6. Create a Hadamard product argument $\left(\pi_1^\times, \widehat{\pi}_1^\times\right)$ for $\left[\!\left[\left(B^\dagger, \widehat{B}^\dagger\right)\right]\!\right] = \left[\!\left[\left(\prod_{j=0}^{\ell_v-1}\left(B'_j\right)^{G'_j}, \prod_{j=0}^{\ell_v-1}\left(\widehat{B}'_j\right)^{G'_j}\right)\right]\!\right] \circ \left[\!\left[\left(\text{Com}^1\left(\widehat{\text{crs}}_1; G_0, \ldots, G_{\ell-1}; 0\right), \prod_{i=0}^{\ell-1} g_{2,\lambda_i}^{G_i}\right)\right]\!\right]$.
7. Create Lipmaa permutation argument $\left(A^*, \widehat{A}^*, \pi_2^\times, \widehat{\pi}_2^\times, \pi_2^{\text{rot}}, \widehat{\pi}_2^{\text{rot}}\right)$ for $\text{rot}\left(\left[\!\left[\left(C, \widetilde{C}\right)\right]\!\right]\right) = \left[\!\left[\left(C_{\text{rot}}, \widehat{C}_{\text{rot}}, \widetilde{C}_{\text{rot}}\right)\right]\!\right]$.
8. Create a Hadamard product argument $\left(\pi_3^\times, \widehat{\pi}_3^\times\right)$ for $\left[\!\left[\left(C/B^\dagger, \widehat{C}/\widehat{B}^\dagger\right)\right]\!\right] = \left[\!\left[\left(C_{\text{rot}}, \widehat{C}_{\text{rot}}\right)\right]\!\right] \circ \left[\!\left[\left(\text{Com}^1\left(\widehat{\text{crs}}_1; 1, 1, \ldots, 1, 0; 0\right), \prod_{i=0}^{\ell-2} g_{2,\lambda_i}\right)\right]\!\right]$.
9. Create a Hadamard product argument $\left(\pi_4^\times, \widehat{\pi}_4^\times\right)$ for $\left[\!\left[\left(A_c^{u-1}, \widehat{A}_c^{u-1}\right)\right]\!\right] = \left[\!\left[\left(C, \widehat{C}\right)\right]\!\right] \circ \left[\!\left[\left(\text{Com}^1\left(\widehat{\text{crs}}_1; 1, 0, \ldots, 0, 0; 0\right), g_{2,\lambda_0}\right)\right]\!\right]$.
10. Create an equality subargument π_5^{ce} that A_c commits to the same value that $\left(A_g, A_f, A_h\right)$ encrypts, using the argument from Section 5.4.
11. Send to the verifier the argument:

$$\pi \leftarrow \left(\begin{array}{l}\left(B'_j, \widehat{B}'_j, B'_{j,2}, \pi'_j, \widehat{\pi}'_j\right)_{j\in\mathbb{Z}_{\ell_v}}, \left(B^\dagger, \widehat{B}^\dagger\right), \left(C, \widehat{C}, \widetilde{C}\right), \left(C_{\text{rot}}, \widehat{C}_{\text{rot}}, \widetilde{C}_{\text{rot}}\right), \\ \left(\pi_1^\times, \widehat{\pi}_1^\times\right), \left(A^*, \widehat{A}^*, \pi_2^\times, \widehat{\pi}_2^\times, \pi_2^{\text{rot}}, \widehat{\pi}_2^{\text{rot}}\right), \left(\pi_3^\times, \widehat{\pi}_3^\times\right), \left(\pi_4^\times, \widehat{\pi}_4^\times\right), \pi_5^{ce}.\end{array}\right)$$

Protocol 5.2b – Argument of the non-interactive range proof protocol for the range $[0, H]$

Verification $\big(\mathsf{crs};(\mathsf{pk},A_g,A_f,A_h,A_c,\widehat{A}_c),\pi\big)$**:** The verifier does the following:

1. For $j\in\mathbb{Z}_{\ell_v}$:
   (a) Check that $e(B'_j,g_2)=e(g_1,B'_{j,2})$ and $e(B'_j,\widehat{g}_2)=e(\widehat{B}'_j,g_2)$.
   (b) Verify the Hadamard product argument $(\pi'_j,\widehat{\pi}'_j)$ for the corresponding inputs.
2. For $K\in\{A_c,B^\dagger,C,C_{\mathsf{rot}}\}$: check that $e(K,\widehat{g}_2)=e(\widehat{K},g_2)$.
3. For $K\in\{C,C_{\mathsf{rot}}\}$: check that $e(K,\widetilde{g}_2)=e(\widetilde{K},g_2)$.
4. Verify the Hadamard product arguments $(\pi_1^\times,\widehat{\pi}_1^\times),(\pi_3^\times,\widehat{\pi}_3^\times),(\pi_4^\times,\widehat{\pi}_4^\times)$, the Lipmaa permutation argument $(A^*,\widehat{A}^*,\pi_2^\times,\widehat{\pi}_2^\times,\pi_2^{\mathsf{rot}},\widehat{\pi}_2^{\mathsf{rot}})$, and the equality subargument π_5^{ce} for the corresponding inputs.

Protocol 5.2c – Verification of the non-interactive range proof protocol for the range $[0,H]$

Protocol explanation. The NIZK range proof of this section is detailed in Protocols 5.2a, 5.2b, and 5.2c. Its basic idea is explained hereinafter. The common input for both parties is equal to a lifted BBS encryption (A_g,A_f,A_h) of σ, accompanied by a knowledge component $\widehat{A}$ such that $(A,\widehat{A})$ is at the same time a knowledge commitment to σ. In the setup of the protocol, $u>1$ has to be chosen according to the communication objectives. A short proof will impose a large u and a large CRS. If the CRS needs to be small, which is obtained with a small u, this will lead to a longer proof. After a $u>1$ has been chosen, let $\ell=\ell(u,(u-1)H)$ be defined as in Section 4.4. According to Theorem 4.10, $\sigma\in[0,H]$ if and only if for G_i computed from equations 4.6 and 4.7 one has $(u-1)\sigma=\sum_{i=0}^{\ell-1}G_i\sigma_i$ for some $\sigma_i\in\mathbb{Z}_u$. Thus, the first step of the prover is to decompose σ into σ_i. Then the prover shows by using parallel versions of the range proof from [LAN02], that for $i\in\mathbb{Z}_\ell$, $\sigma_i\in\mathbb{Z}_u$. Note that the range proof in [LAN02] is the binary version of the sumset based range proof presented in Section 4.5. Showing that $\sigma_i\in\mathbb{Z}_u$ is done by writing σ_i as $\sigma_i=\sum_{j=0}^{\ell_v-1}G'_j\sigma'_{j,i}$, where $\ell_v=\lfloor\log_2(u-1)\rfloor$, $G'_j=\lfloor(u-1+2^j)/2^{j+1}\rfloor$, for some $\sigma'_{j,i}\in\{0,1\}$. The latter results from the binary case of Theorem 4.10 and from Corollary 4.12. Showing that $\sigma'_{j,i}\in\{0,1\}$ is achieved by using a Hadamard product argument. This product argument will be performed on commitments on $(\sigma'_{j,0},\ldots,\sigma'_{j,\ell-1})$ for $j\in\mathbb{Z}_{\ell_v}$. These last commitments will be informally denoted B'_j.

The prover then commits to the vector $\boldsymbol{c}=(c_0,\ldots,c_{\ell-1})$, where $c_j=\sum_{i=j}^{\ell-1}G_i\sigma_i$, and shows that the values c_j are correctly computed by using a small constant number of Hadamard product and Lipmaa permutation arguments. Moreover (and informally), the prover first computes the values c_j in the forth step of the protocol, then he creates in the fifth step, the commitments $B^\dagger$ on $(G_0\sigma_0,\ldots,G_{\ell-1}\sigma_{\ell-1})$, C on $(c_0,\ldots,c_{\ell-1})$, and C_{rot} on $(c_1,\ldots,c_{\ell-1},c_0)$. The sixth step of the protocol uses a Hadamard product argument to show that $B^\dagger$ has been correctly formed, from G'_j and from the commitments B'_j. The seventh step uses a permutation argument to show that C_{rot} is a correct rotation permutation by one element of C. Then, in the eighth step, the prover

uses a Hadamard product argument to show that $c_{j+1} = c_j - G_j\sigma_j$, with $c_{\ell-1} = G_{\ell-1}\sigma_{\ell-1}$. This latter argument is obtained from the commitment on $(c_1,\ldots,c_{\ell-1},0)$, which is derived from C_{rot}. It shows that:

$$(c_1,\ldots,c_{\ell-1},0) = (c_0,\ldots,c_{\ell-1}) - (G_0\sigma_0,\ldots,G_{\ell-1}\sigma_{\ell-1}).$$

Thus, the verifier will be convinced that $c_j = \sum_{i=j}^{\ell-1} G_i\sigma_i$. But then, by Theorem 4.10, $c_0 = \sum_{i=0}^{\ell-1} G_i\sigma_i \in [0,(u-1)H]$. After this step, the prover shows, using a single Hadamard product argument, that $(A_c^{u-1}, \widehat{A}_c^{u-1})$ commits to $(c_0,0,\ldots,0)$. This will imply that the secret σ committed in $(A_c,\widehat{A}_c)$ is indeed in the range $[0,H]$. The last required step links the secret σ contained in the knowledge commitment $(A_c,\widehat{A}_c)$ to the lifted BBS encryption (A_g,A_f,A_h) of σ with randomizers (r_f,r_h), where $r = r_f + r_h$. This last step is achieved with the subargument explained in Section 5.4.

As in [Lip12a], in a few cases, instead of computing two different commitments $\mathrm{Com}^t(\widehat{\mathsf{crs}}_t;\boldsymbol{a};r) = (g_t^r\cdot\prod g_{t,\lambda_i}^{a_i}, \widehat{g}_t^r\cdot\prod \widehat{g}_{t\lambda_i}^{a_i})$ and $\mathrm{Com}^t(\widetilde{\mathsf{crs}}_t;\boldsymbol{a};r) = (g_t^r\cdot\prod g_{t,\lambda_i}^{a_i}, \widetilde{g}_t^r\cdot\prod \widetilde{g}_{t,\lambda_i}^{a_i})$, the following composed commitment is computed:

$$\mathrm{Com}^t(\mathsf{crs}_t;\boldsymbol{a};r) = (g_t^r\cdot\prod g_{t,\lambda_i}^{a_i}, \widehat{g}_t^r\prod \widehat{g}_{t,\lambda_i}^{a_i}, \widetilde{g}_t^r\cdot\prod \widetilde{g}_{t,\lambda_i}^{a_i}).$$

Theorem 5.2

Let $u > 1$. Let $H = poly(\kappa)$ and $\ell = \ell(u,(u-1)H)$ be defined as in Section 4.4. Let $\Lambda = \{\lambda_i\}_{i\in\mathbb{Z}_\ell} \subset \mathbb{N}$ be such that $\forall i < j\colon 0 < \lambda_i < \lambda_j$. Let $\widehat{\Lambda} := \{0\}\cup\Lambda\cup 2\widehat{}\Lambda$, and $\widetilde{\Lambda}$ as in Protocol 5.2a. Let rot be a permutation from $\mathbb{Z}_\ell$ to $\mathbb{Z}_\ell$, where $\mathsf{rot}(i) = i-1$ if $i > 0$, and $\mathsf{rot}(0) = \ell - 1$. Define G_i with equations 4.6 and 4.7. The argument detailed by Protocols 5.2a, 5.2b, and 5.2c, is perfectly complete. If the asymmetric bilinear group generator PG_a is Λ-PKE secure and DLIN secure in $\mathbb{G}_1$, then the argument detailed by Protocols 5.2a, 5.2b, and 5.2c, is computationally zero-knowledge. If PG_a is $\widetilde{\Lambda}$-PSDL secure and Λ-PKE secure in both $\mathbb{G}_1$ and $\mathbb{G}_2$, then the argument detailed by Protocols 5.2a, 5.2b, and 5.2c, is computationally sound.

Proof

To show that the argument described by Protocols 5.2a, 5.2b, and 5.2c, is a NIZK argument, three security properties are proven: *perfect completeness*, *computational soundness*, and *computational zero-knowledge*.

PERFECT COMPLETENESS: Recall that in the case of the product arguments, the inputs of the prover are $(A,\widehat{A},B,\widehat{B},B_2,C,\widehat{C})$. Within this proof, it is presumed that $(B,\widehat{B},B_2)$ (assuming B_2 is correctly defined, that is, $e(B,g_2) = e(g_1,B_2)$) commits to the same values as $(B,\widehat{B})$.

The pairing verifications (for example, that $e(K,\widehat{g}_2) = e(\widehat{K},g_2)$) hold by construction of the protocol. Since $(B'_j,\widehat{B}'_j)$ commits to $(\sigma'_{j,0},\ldots,\sigma'_{j,\ell-1})$ for binary $\sigma'_{j,i}$ then the argument $(\pi'_j,\widehat{\pi}'_j)$ verifies.

Note that $\left(\prod_{j=0}^{\ell_v-1}(B'_j)^{G'_j}, \prod_{j=0}^{\ell_v-1}(\widehat{B}'_j)^{G'_j}\right)$ commits to $(\sigma_0,\ldots,\sigma_{\ell-1})$. Thus argument $(\pi_1^\times,\widehat{\pi}_1^\times)$ verifies. Since $(C_{\mathsf{rot}},\widehat{C}_{\mathsf{rot}})$ commits to a rotation of $(C,\widehat{C})$, then $(A^*,\widehat{A}^*,\pi_2^\times,\widehat{\pi}_2^\times,\pi_2^{\mathsf{rot}},\widehat{\pi}_2^{\mathsf{rot}})$ verifies.

Since $(C_{\mathsf{rot}}, \widehat{C}_{\mathsf{rot}})$ commits to $(c_0, \ldots, c_{\ell-1}, 0)$ and $(C/B^{\dagger}, \widehat{C}/\widehat{B}^{\dagger})$ commits to

$$(c_0 - G_0\sigma_0, c_1 - G_1\sigma_1, \ldots, c_{\ell-1} - G_{\ell-1}\sigma_{\ell-1}) = (c_0, \ldots, c_{\ell-1}, 0),$$

then $(\pi_3^{\times}, \widehat{\pi}_3^{\times})$ verifies. Finally, since $(u-1)\sigma = \sum_{i=0}^{\ell-1} G_i\sigma_i$ and $c_0 = \sum_{i=0}^{\ell-1} G_i\sigma_i$, then $(\pi_4^{\times}, \widehat{\pi}_4^{\times})$ verifies.

COMPUTATIONAL SOUNDNESS: let $\mathscr{A}$ be a non-uniform PPT adversary who creates a statement $(\mathsf{pk}, A_g, A_f, A_h, A_c, \widehat{A}_c)$ and an accepting range proof π. By the DLIN assumption, the lifted BBS cryptosystem is IND-CPA secure, and thus the adversary obtains no information from (A_g, A_f, A_h). By the Λ-PKE assumption, there exists a non-uniform PPT extractor $X_{\mathscr{A}}$ that, running on the same inputs and seeing $\mathscr{A}$'s random tape, extracts the following openings:

$$\begin{aligned}
\left(A_c,\ \widehat{A}_c\right) &= \left(g_1^{\tilde{r}} g_{1,\lambda_0}^{\sigma},\quad \widehat{g}_1^{\tilde{r}} \widehat{g}_{1,\lambda_0}^{\sigma}\right),\\
\left(B'_j,\ \widehat{B}'_j\right) &= \mathrm{Com}^1\left(\widehat{\mathsf{crs}}_1; \boldsymbol{b'_j}; r_j\right) \text{ for } j \in \mathbb{Z}_{\ell_v},\\
\left(B^{\dagger},\ \widehat{B}^{\dagger}\right) &= \mathrm{Com}^1\left(\widehat{\mathsf{crs}}_1; \boldsymbol{b}^{\dagger}; r'_0\right),\\
\left(C,\ \widehat{C}\right) &= \mathrm{Com}^1\left(\widehat{\mathsf{crs}}_1; \boldsymbol{c}; r'_1\right),\\
\left(C_{\mathsf{rot}},\ \widehat{C}_{\mathsf{rot}}\right) &= \mathrm{Com}^1\left(\widehat{\mathsf{crs}}_1; \boldsymbol{c_{\mathsf{rot}}}; r'_2\right),\\
\left(\pi_1^{\times},\ \widehat{\pi}_1^{\times}\right) &= \left(\prod_{s\in\widehat{\Lambda}} g_{2,s}^{f'_{(\times 1,s)}},\quad \prod_{s\in\widehat{\Lambda}} \widehat{g}_{2,s}^{f'_{(\times 1,s)}}\right),\\
\left(A^*,\ \widehat{A}^*\right) &= \mathrm{Com}^1\left(\widehat{\mathsf{crs}}_1; \boldsymbol{a}^*; r_{a^*}\right),\\
\left(\pi_2^{\times},\ \widehat{\pi}_2^{\times}\right) &= \left(\prod_{s\in\widehat{\Lambda}} g_{2,s}^{f'_{(\times 2,s)}},\quad \prod_{s\in\widehat{\Lambda}} \widehat{g}_{2,s}^{f'_{(\times 2,s)}}\right),\\
\left(\pi_2^{\mathsf{rot}},\ \widehat{\pi}_2^{\mathsf{rot}}\right) &= \left(\prod_{s\in\widetilde{\Lambda}} g_{2,s}^{f'_{(\mathsf{rot}2,s)}},\quad \prod_{s\in\widetilde{\Lambda}} \widetilde{g}_{2,s}^{f'_{(\mathsf{rot}2,s)}}\right),\\
\left(\pi_3^{\times},\ \widehat{\pi}_3^{\times}\right) &= \left(\prod_{s\in\widehat{\Lambda}} g_{2,s}^{f'_{(\times 3,s)}},\quad \prod_{s\in\widehat{\Lambda}} \widehat{g}_{2,s}^{f'_{(\times 3,s)}}\right), \text{ and}\\
\left(\pi_4^{\times},\ \widehat{\pi}_4^{\times}\right) &= \left(\prod_{s\in\widehat{\Lambda}} g_{2,s}^{f'_{(\times 4,s)}},\quad \prod_{s\in\widehat{\Lambda}} \widehat{g}_{2,s}^{f'_{(\times 4,s)}}\right).
\end{aligned}$$

The extractor $X_{\mathscr{A}}$ will also create the openings that correspond to π_5^{ce}. Since the $\widetilde{\Lambda}$-PSDL assumption is supposed to hold, all the following is true. In the contrary (in the case that it is not true), one can efficiently test it, and thus break the PSDL assumption.

Since $e(B'_j, g_2) = e(g_1, B'_{j,2})$ for $j \in \mathbb{Z}_{\ell_v}$, then $(B'_j, \widehat{B}'_j, B_{j,2})$ commits to $\boldsymbol{b'_j}$. Therefore, due to the $\widehat{\Lambda}$-PSDL assumption, Theorem 2.3, the fact that the adversary knows the openings of $\left(\left(B'_j, \widehat{B}'_j\right), \left(\pi'_j, \widehat{\pi}'_j\right)\right)$, and since $(\pi'_j, \widehat{\pi}'_j)$ verifies, then $\sigma'_{j,i} \in \{0,1\}$ for all $j \in \mathbb{Z}_{\ell_v}$ and $i \in \mathbb{Z}_{\ell}$. Thus, by Theorem 4.10, $\boldsymbol{b} = (\sigma_0, \ldots, \sigma_{\ell-1}) := (\sum_{j=0}^{\ell_v-1} G'_j\sigma'_{j,0}, \ldots, \sum_{j=0}^{\ell_v-1} G'_j\sigma'_{j,n}) \in \mathbb{Z}_u^{\ell}$, and thus $(\prod_{j=0}^{\ell_v-1} (B'_j)^{G'_j}, \prod_{j=0}^{\ell_v-1} (\widehat{B}'_j)^{G'_j})$ commits to $\boldsymbol{b}$ with $\sigma_i \in \mathbb{Z}_u$.

Due to the $\widehat{\Lambda}$-PSDL assumption, Theorem 2.3, the fact that the adversary knows the openings of $\left(\left(B'_j, \widehat{B}'_j\right), \left(B^\dagger, \widehat{B}^\dagger\right), \left(\pi_1^\times, \widehat{\pi}_1^\times\right)\right)$, and since $\left(\pi_1^\times, \widehat{\pi}_1^\times\right)$ verifies, then $\sigma_i^\dagger = G_i \sigma_i$.

Due to the $\widetilde{\Lambda}$-PSDL assumption, Theorem 2.3, Theorem 2.5, the fact that the adversary knows the openings of $\left(\left(C, \widetilde{C}\right), \left(C_{\mathsf{rot}}, \widehat{C}_{\mathsf{rot}}\right), \left(A^*, \widehat{A}^*, \pi_2^\times, \widehat{\pi}_2^\times, \pi_2^{\mathsf{rot}}, \widehat{\pi}_2^{\mathsf{rot}}\right)\right)$, and since $\left(A^*, \widehat{A}^*, \pi_2^\times, \widehat{\pi}_2^\times, \pi_2^{\mathsf{rot}}, \widehat{\pi}_2^{\mathsf{rot}}\right)$ verifies, then $c_{\mathsf{rot},\ell-1} = c_0$ and $c_{\mathsf{rot},i-1} = c_i$ for $\ell - 1 \geqslant i \geqslant 1$.

Due to the $\widehat{\Lambda}$-PSDL assumption,Theorem 2.3, the fact that the adversary knows the openings of $\left(\left(C_{\mathsf{rot}}, \widehat{C}_{\mathsf{rot}}\right), \left(C, \widehat{C}\right), \left(B^\dagger, \widehat{B}^\dagger\right), \left(\pi_3^\times, \widehat{\pi}_3^\times\right)\right)$, and since $\left(\pi_3^\times, \widehat{\pi}_3^\times\right)$ verifies, then $c_{\ell-1} - G_{\ell-1}\sigma_{\ell-1} = c_{\mathsf{rot},\ell-1} = 0$ and $c_i - G_i\sigma_i = c_{\mathsf{rot},i} = c_{i+1}$ for $\ell - 1 > i \geqslant 0$. Therefore, $c_{\ell-1} = G_{\ell-1}\sigma_{\ell-1}$, $c_{\ell-2} = G_{\ell-2}\sigma_{\ell-2} + G_{\ell-1}\sigma_{\ell-1}$, and by induction $c_i = \sum_{j=i}^{\ell-1} G_j\sigma_j$ for $i \in \mathbb{Z}_\ell$. This implies that $c_0 = \sum_{i=0}^{\ell-1} G_i\sigma_i$ for $\sigma_i \in \mathbb{Z}_u$.

Due to the $\widehat{\Lambda}$-PSDL assumption, Theorem 2.3, the fact that the adversary knows the openings of $\left(\left(C, \widehat{C}\right), \left(A_c, \widehat{A}_c\right), \left(\pi_4^\times, \widehat{\pi}_4^\times\right)\right)$, and since $\left(\pi_4^\times, \widehat{\pi}_4^\times\right)$ verifies, then $\left(A_c, \widehat{A}_c\right) = \left(g_1^{\tilde{r}} g_{1,\lambda_0}^{\sigma}, \widehat{g}_1^{\tilde{r}} \widehat{g}_{1,\lambda_0}^{\sigma}\right)$ commits to $(\sigma, 0, \ldots, 0)$ such that $(u-1)\sigma = \sum_{i=0}^{\ell-1} G_i\sigma_i$ for $\sigma_i \in \mathbb{Z}_u$, and therefore by Theorem 4.10, $\sigma \in [0, H]$.

Due to the $\{\lambda_0\}$-PSDL assumption and since π_5^{ce} verifies, then $\left(A_g, A_f, A_h\right)$ encrypts $\sigma \in [0, H]$.

COMPUTATIONAL ZERO-KNOWLEDGE: to prove computational zero-knowledge, the following simulator $Sim = (Sim_1, Sim_2)$ is constructed. Firstly, Sim_1 creates a correctly formed common reference string together with a simulation trapdoor $\mathsf{td} = \left(\widehat{\alpha}, \widetilde{\alpha}, \alpha_g, \alpha_f, \alpha_h, \alpha_{g/c}, x\right)$. After that, the prover creates a statement $input^{\tilde{r}} := \left(\mathsf{pk}, A_g, A_f, A_h, A_c, \widehat{A}_c\right)$ and sends it to the simulator Sim. Secondly, $Sim_2\left(\mathsf{crs}; input^{\tilde{r}}; \mathsf{td}\right)$ uses a knowledge extractor to extract $(\boldsymbol{a}, \tilde{r})$ from the random coins of the prover and $\left(A_c, \widehat{A}_c\right)$. The goal of the simulator Sim is to simulate the argument of an honest prover. Therefore, the statement $input^{\tilde{r}}$ is considered to have been generated by an honest prover. This implies that $\boldsymbol{a} = (\sigma, 0, \ldots, 0)$ with $\sigma \in [0, H]$. Thus, using the fact that the knowledge commitment scheme is also trapdoor, the simulator computes $r'' \leftarrow \sigma x^{\lambda_0} + \tilde{r}$, which implies the equality $A_c = g_1^{r''}$. Since both $\tilde{r}$ and r'' are uniformly random, r'' does not leak any information on the input of the prover. Thereafter, the simulator creates all commitments $\left(B'_j, \widehat{B}'_j, B'_{j,2}\right)_{j \in \mathbb{Z}_{\ell_v}}$, $\left(B^\dagger, \widehat{B}^\dagger\right)$, $\left(C, \widehat{C}, \widetilde{C}\right)$, and $\left(C_{\mathsf{rot}}, \widehat{C}_{\mathsf{rot}}, \widetilde{C}_{\mathsf{rot}}\right)$ as in the argument, but replacing $\boldsymbol{a}$ with 0 and $\tilde{r}$ with r''. Therefore, all of the aforementioned commitments just commit to $\mathbf{0}$. Thus, the simulator can simulate all product and permutation arguments, the equality subargument of Section 5.4, and form the general simulated argument π^{sim}. Clearly, this simulated argument π^{sim} is perfectly indistinguishable from the real argument π.

Note that the use of a cryptosystem makes achieving perfect zero-knowledge impossible. Furthermore, $(A_c, \widehat{A}_c)$ is provided by the prover and not generated during the argument. To achieve zero-knowledge, one must be able to open $(A_c, \widehat{A})$ having been given only the CRS trapdoor. That is, one has to use an extractable commitment scheme [Cre02, ACP09]. It is easy to see that the knowledge commitment scheme is extractable, however, extractability is only achieved under the PKE assumption. ■

Theorem 5.3

Let $u > 1$. Let Λ be defined as in Theorem 2.1 and let $\ell = \ell(u,(u-1)H) \leqslant 1 - \log_u 2 + \log_u((u-1)H-(u-2)) \leqslant \lceil \log_u(H) \rceil$, where $\ell(\cdot,\cdot)$ is defined as in Section 4.4.
Let $\ell_v = \lfloor \log_2(u-1) \rfloor$. Assume that the Hadamard product argument from Section 2.4.7 and the Lipmaa permutation argument from Section 2.4.8 are used. The range proof described with Protocols 5.2a, 5.2b, and 5.2c, has a common reference string of length $\ell^{1+o(1)}$ elements, a communication complexity of $2\ell_v + 21$ elements from $\mathbb{G}_1$ and $3\ell_v + 14$ elements from $\mathbb{G}_2$. The computational complexity of its prover is dominated by $\Theta(\ell^2 \ell_v)$ scalar multiplications in $\mathbb{Z}_p$ and $\ell_v \cdot \ell^{1+o(1)}$ exponentiations (in $\mathbb{G}_1$ or $\mathbb{G}_2$). The computational complexity of its verifier is dominated by $9\ell_v + 72$ pairings.

Proof

The CRS is composed of: the parameters of the asymmetric bilinear groups $(\mathsf{param}_{\mathsf{bp}})$, $(9+3|\Lambda|)$ group elements in $\mathbb{G}_1$, and $\left(10+\left|\widehat{\Lambda}\right|+2\left|\widetilde{\Lambda}\right|\right)$ group elements in $\mathbb{G}_2$. As the following hold,

$$\begin{aligned}
|\Lambda| &= \ell, \\
\left|\widehat{\Lambda}\right| &= 1+\ell+\ell(\ell-1) \\
&= \ell^2+1, \\
\left|\widetilde{\Lambda}\right| &\leqslant \ell+\ell^2+\ell(\ell-1)+\ell^2(\ell-1) \\
&\leqslant \ell^3-\ell^2,
\end{aligned}$$

the CRS length is composed of: $(\mathsf{param}_{\mathsf{bp}})$, $(9+3\ell)$ group elements in $\mathbb{G}_1$, and less than $\left(2\ell^3-\ell^2+11\right)$ group elements in $\mathbb{G}_2$. Therefore, the CRS length is $\ell^{1+o(1)}$ elements.

The communication complexity is composed of:

- ℓ_v tuples $(B'_j, \hat{B}'_j, B'_{j2}, \pi'_j, \widehat{\pi}'_j)$,
  where each tuple has 2 elements of $\mathbb{G}_1$ and 3 elements of $\mathbb{G}_2$,
- 8 extra elements from $\mathbb{G}_1$,
- 3 Hadamard product arguments, where each argument has 2 elements from $\mathbb{G}_2$,
- the permutation argument, which has 2 elements from $\mathbb{G}_1$ and 4 elements from $\mathbb{G}_2$,
- and the equality subargument π^{ce},
  which has 11 elements from $\mathbb{G}_1$ and 4 elements from $\mathbb{G}_2$.

Thus, in total, the communication complexity is of $2\ell_v + 8 + 2 + 11 = 2\ell_v + 21$ elements from $\mathbb{G}_1$ and $3\ell_v + 3\cdot 2 + 4 + 4 = 3\ell_v + 14$ elements from $\mathbb{G}_2$.

The computational complexity of the prover is dominated by $\ell_v + 3$ Hadamard product arguments ($\Theta(\ell^2)$ scalar multiplications in $\mathbb{Z}_p$ and $\ell^{1+o(1)}$ exponentiations in bilinear groups each), by the permutation argument ($\Theta(\ell^2)$ scalar additions in $\mathbb{Z}_p$ and $\ell^{1+o(1)}$ exponentiations in bilinear groups), and by the equality subargument (13 exponentiations in $\mathbb{G}_1$ and 8 exponentiations in $\mathbb{G}_2$). In total, the computational complexity of the prover is thus domi-

nated by $\Theta(\ell^2 \cdot \ell_v) = \Theta(\ell^2 \cdot \log u)$ scalar multiplications in $\mathbb{Z}_p$ and $\ell_v \cdot \ell^{1+o(1)} = \ell^{1+o(1)} \cdot \log u$ exponentiations in bilinear groups.

The computational complexity of the verifier is dominated by verifying $\ell_v + 3$ Hadamard product arguments (5 pairings each), the permutation argument (12 pairings), and the equality subargument π^{ce} (33 pairings). In addition, the verifier performs $4\ell_v + 4 \cdot 2 + 2 \cdot 2 = 4\ell_v + 12$ pairings. The total number of pairings is thus $9\ell_v + 72$. ■

The communication complexity is minimized when ℓ_v (and thus u) is as small as possible, that is, $u = 2$. Then $\ell_v = \lfloor \log_2 1 \rfloor = 0$. In this case the communication consists of 21 elements from $\mathbb{G}_1$ and 14 elements from $\mathbb{G}_2$. The same choice $u = 2$ is also optimal for the computational complexity of the verifier (72 pairings). As noted before in Section 5.4, at the security level of 2^{128}, elements of $\mathbb{G}_1$ can be represented in 256 bits, and elements of $\mathbb{G}_2$ in 512 bits. Thus, at this security level, if $u = 2$ then the communication is $21 \cdot 256 + 14 \cdot 512 = 12\,544$ bits. Therefore, the communication complexity is even smaller than that of positivity testing based arguments like [Bou00, Lip03, Gro05, Sce09].

The optimal computational complexity for the prover is achieved when the number of exponentiations, $\ell^{1+o(1)} \cdot \ell_v = (\log_u H)^{1+o(1)} \cdot \lfloor \log_2(u-1) \rfloor$, is minimized. This happens when $u = H$. The computation of the prover is then dominated by $\Theta(\log H)$ scalar multiplications and exponentiations. Moreover, in this case the CRS length $\ell^{1+o(1)}$ is constant.

Finally,the summatory length of the CRS and the communication may be required to be minimal, that is, $\ell^{1+o(1)} + \Theta(\ell_v)$. Considering $\ell \leqslant \log_u H$ and $\ell_v \leqslant \log_2 u$, the sum becomes $(\log_u H)^{1+o(1)} + \Theta(\log u)$. This sum can be approximately minimized by choosing $u = 2^{\sqrt{\log H}}$. Then the summatory length becomes $\left(\log H\right)^{1/2+o(1)}$. In this case, it would make sense to change the role of groups $\mathbb{G}_1$ and $\mathbb{G}_2$ to get better efficiency. The efficiency of the lifted BBS encryption based non-interactive range proof (Protocols 5.2a, 5.2b, and 5.2c) in all three cases, is given in Figure 5.1.

	CRS length	Argument π length	Prover computations	Verifier computations
[RKP09]	$\Theta(1)$	$\Theta(\|H\|)$	$\Theta(\|H\|)$ E	$\Theta(\|H\|)$ P
[RKP09]	$\Theta\left(\frac{\|H\|}{\log\|H\|}\right)$	$\Theta\left(\frac{\|H\|}{\log\|H\|}\right)$	$\Theta\left(\frac{\|H\|}{\log\|H\|}\right)$ E	$\Theta\left(\frac{\|H\|}{\log\|H\|}\right)$ P
Lifted BBS Encryption Based Non-Interactive Range Proof (Protocols 5.2a, 5.2b, and 5.2c)				
General	$\ell^{1+o(1)}$	$5\ell_v+35$	$\Theta\left(\ell_v\cdot\ell^2\right)$M $+\Theta\left(\ell_v\cdot\ell^{1+o(1)}\right)$E	$(9\ell_v+72)$ P
$u=2$	$\|H\|^{1+o(1)}$	35	$\Theta\left(\|H\|^2\right)$M $+\|H\|^{1+o(1)}$E	72 P
$u=2^{\sqrt{\|H\|}}$	$\|H\|^{1/2+o(1)}$	$\approx 5\sqrt{\|H\|}+35$	$\Theta\left(\|H\|^{3/2}\right)$M $+\|H\|^{1+o(1)}$E	$\approx\left(9\sqrt{\|H\|}+72\right)$ P
$u=H$	$\Theta(1)$	$\approx 5\|H\|+35$	$\Theta(\|H\|)$M $+\Theta(\|H\|)$E	$\approx(9\|H\|+72)$ P

Figure 5.1 – Comparison of NIZK arguments for range proof.
Here, M/E/P means the number of multiplications, exponentiations and pairings respectively. Communication is given in group elements. Furthermore, $\ell_v=\lfloor\log(u-1)\rfloor$, $\ell\approx\log_u H$, and the basis of all logarithms is 2. Recall that $\|H\|=\log_2 H$.

Part II

Extended Access Control

Chapter 6

Machine Readable Travel Documents

In this chapter, which is with minor revisions based on sections of [CV09] and [Cha13], the Machine Readable Travel Document (MRTD) standard evolution is surveyed and the remaining problems explained. The next chapter will propose directions and solutions for the next upgrades in order to suppress these problems. In Section 6.2 an overview will be provided of prior and related work. Section 6.3 will then explain and give the drawbacks of the Radio Frequency IDentification (RFID). The International Civil Aviation Organization (ICAO) standard will be explained in Section 6.4, and the Extended Access Control (EAC) version 1 (EACv1) and version 2 (EACv2) respectively in Sections 6.5 and 6.6. Conclusions will be provided in Section 6.7.

6.1 Introduction

Since 2004, a majority of countries have adopted the ICAO standard [ICAO04a, ICAO04b] for Machine Readable Travel Documents (MRTDs). Among other things, this standard specifies how to store and use biometrics in passports in order to have more secure identification of the holder. Since it is based on the RFID technology [ISO10a], an access control is necessary for privacy protection. The optional one proposed in the ICAO standard is based on symmetric-key cryptography with a key printed on the passport. It is called Basic Access Control (BAC), offers very little privacy protection, and is the only mechanism which can be used to protect mandatory data groups (DGs) containing the identifiers of holders.

In response to the initial weak standard for MRTDs produced by the ICAO, the European Union has mandated the Federal Office for Information Security (BSI) to provide and maintain a stronger standard for MRTDs. In that regard, the BSI has issued the Extended Access Control (EAC) which provides a stronger privacy protection for MRTDs. Its first initial release, EACv1 [BSI06], was made in 2006 to have a reasonably secure privacy protection for other data groups. It is based on public-key cryptography and requires a public key infrastructure to be

deployed for readers. Since passports are not online, they cannot receive certificate revocation lists. Thus, revocation can only be based on expiration dates. Unfortunately, passports do not have a clock, so they can only compare the validity period with the latest accepted certificate date. EACv1 protects against cloning but only where it is being used in a country with the ability to read EAC compatible MRTDs. Although not mandatory, countries with the ability to read EACv1 compatible MRTDs but being unauthorized to pass terminal authentication, could use privacy-enhanced protocols.

The second and latest version EACv2 [BSI15a, BSI15b, BSI15c, BSI15d] was introduced in 2009 and corrected in 2012 and in 2015. EACv2 makes sure that passports are only read by authorized terminals, which puts an end to the cloning issue. Indeed, EACv2 goes further by protecting access to ICAO-mandatory data groups, even for countries unauthorized to read other data groups. It was believed that with the introduction of EACv2 in 2009, the majority of threats were solved. Unfortunately, ICAO-mandatory data groups must be readable by countries not implementing EAC so this protocol is likely to be bypassed for interoperability reasons. Furthermore, several flaws and threats remain. The major flaw that can be pointed out is the absence of a good terminal revocation. The other issues are now considered marginal as they are or will gradually be solved with the evolution of previous standards (notably the one from the ICAO [ICAO08, ICAO13]). However, no progress has been made regarding terminal revocation nor with regard to terminal authentication.

6.2 Prior and Related work

A substantial amount of work has already been achieved on MRTDs. Juels, Molnar, and Wagner [JMW05] presented one of the first (if not the first) security analysis on e-passports in 2005. They identified several flaws in the ICAO standard, namely clandestine scanning, clandestine tracking, skimming then cloning, eavesdropping, biometric data-leakage, and weaknesses in the cryptographic setups of the ICAO standard. Kc and Karger [KK05] presented their research on similar tracks in 2005 and introduced some other attacks, namely the “splicing” attack and the “fake finger” attack. In 2006, Kosmerlj et al. [KFHS06] studied the weakness of facial recognition. Hoepman et al. [HHJ$^+$06] focused in 2006 on passive attacks against the Basic Access Control (BAC) and provided some thoughts on biometrics. They showed that the entropy of the symmetric key used between the reader and the MRTD is less than 80 bits, and can easily be guessed. Regardless of the knowledge of this secret key, they also explained how a MRTD can be traced back to individuals or groups in the classical case of skimming. Hancke [Han06] and Carluccio et al. [CLRPS06] reported experimental attacks against BAC in 2006. Hancke showed a practical eavesdropping together with a relay attack, and Carluccio et al. emphasized the traceability issue of MRTDs. Liu et al. [LKLRP07] explained how to make a passive decryption attack. In 2009, Danev, Heydt-Benjamin, and Čapkun [DHBC09] demonstrated how to identify individual MRTDs through the physical-layer of RFID tags. They explained that this fact can help in the determination of cloned passports whilst on the other hand suppressing location privacy.

In 2007, Hlaváč and Rosa [HR07] studied the case of Active Authentication (AA) and presented a man-in-the-middle cloning attack against AA. AA is also subject to a *challenge semantics* attack as shown in [BSI08a] and explained in section 6.4.

In 2006, Lehtonen et al. [LMSF06] proposed a potential solution for MRTDs. As a necessary optical contact has to be achieved between a reader and the MRTD to retrieve the MRZinfo, they proposed to combine an optical memory device with the actual RFID chip. This would enable the establishment of a secure channel, as a line of sight is necessary. Eavesdropping and skimming will therefore no longer be possible. Herrigel and Zhao [HZ06] proposed to use a digital watermarking technique to increase the seed entropy, which would be readable by optical scanning. However the main disadvantage of these two papers is that a hardware improvement needs to be performed on passports.

Vaudenay and Vuagnoux [VV07] presented a survey on existing protocols for MRTD and their corresponding weaknesses in 2007, namely the ICAO standards (BAC and AA) and the EU standard (EAC). In the same year, Monnerat, Vaudenay, and Vuagnoux [MVV07] focused on the privacy concerns attached to the release of the passport Security Object Document (SOD). The latter leaks the hash of protected data groups and evidence on private data (see also [Vau07]). In 2007, Lekkas and Gritzalis [LG07] worked on the possibility of using the ICAO standard in order to build a globally interoperable Public Key Infrastructure. However they drew negative conclusions due to several issues such as the lack of a passport revocation mechanism. In 2008, Pasupathinathan, Pieprzyk, and Wang [PPW08a, PPW08b, PPW08c] achieved a formal security analysis on the Australian e-passport and identified several flaws in EACv1, after which they proposed an enhanced version called OSEP. They introduced the need to execute terminal authentication before chip authentication. In 2008, Abid and Afifi [AA08] incorporated the use of elliptic curves in OSEP.

All of these studies pushed the *Bundesamt für Sicherheit in der Informationstechnik* (BSI), in charge of the EAC standardization, to present a new version (EACv2) in October 2008 [BSI08b]. Nithyanand [Nit09] released the first survey on EACv2 in 2009. It claimed that EACv2 solved all of the previous problems except one vulnerability. It is possible to use a reader with an expired certificate to read passports whose internal date is outdated. Unfortunately, this is not the only problem left within EACv2. The current version of EAC is Version 2.20 published in February 2015. It is split into four parts [BSI15a, BSI15b, BSI15c, BSI15d] and contains several minor changes compared to the 2.0 version released in 2008 [BSI08b].

6.3 ISO Standard for RFID

In order to discover the RFID tags in proximity, according to the ISO standard for RFID [ISO10a], readers send a discovery signal. Any RFID tag receiving this signal will reply with a specific identifier in order to allow readers to enter in communication with them. For regular RFID tags, this identifier is constant to enable an easy way to track chips. However

this property is not always desirable for tags, especially when location privacy needs to be protected. This is the case for MRTDs. The solution proposed by the ISO standard is to use a session-dependent randomly generated identifier. This solution has been adopted by almost all countries. Unfortunately, there are discrepancies in the way it is implemented [MVV07]. There are other protocol implementation differences such as availability of optional features, lower layer protocols, and speed of transmission, which allow for the identification of a passport nationality [VV07].

It is a well known fact that privacy must be addressed accross all protocol layers [AO05]. As a matter of fact, recent work by Danev et al. [DHBC09] shows that any RFID tag can be accurately identified according to its physical-layer communication properties, namely by some kind of radio fingerprint. Although their work uses this property to enable cloning detection, the straightforward drawback is the tag tracking feature.

Furthermore, the distance to eavesdrop or to interact with RFID tags is highly underestimated. According to an announcement by the Swiss Federal Office of Communication (OFCOM) [BK08] in November 2008, and even though currently commercialized readers can interact only within a few centimeters, it would be possible to access MRTD from far away (up to 25 meters) by changing readers antenna. In addition, it was announced that radio communication between a legitimate reader and a passport induce a signal on the power line that can be captured 500 meters away.

6.4 ICAO Standard and BAC

Following the ICAO standard, passports must provide passive authentication for two mandatory data groups (DGs):

- Data group DG1 is a digital copy of the printed Machine Readable Zone (MRZ) which includes some basic information about the holder: name, nationality, gender, date of birth, as well as passport serial number and expiration date.
- Data Group DG2 is a digital picture of the face which is optimized for automatic face recognition.

Passive authentication is performed by means of the Security Object of the Document (SOD), which is essentially a digital signature on the list of the hash of data groups together with the certificate of the verifying key. This certificate is computed by the issuing country and the root verifying key of the PKI is assumed to be authenticated by special protocols. Following the state of the art in cryptography, digital signatures are unforgeable and identities can no longer be forged maliciously.

Biometric identification is mostly performed by 2D facial recognition, and soon will be by

fingerprint as well. It could use iris recognition but this technology does not appear to be implemented yet. Nevertheless, 2D-facial recognition is fairly weak and fingerprints could be faked. Fake fingerprints can be made using candy [Mat02] or medecine against constipation [BT09].

Passports could limit themselves to providing DG1, DG2, and SOD in a passive way. Indeed, they could be printed using a 2D barcode or a Quick Response (QR) Code, but ICAO preferred RFID-based technology in order to accommodate more data and functionalities in the future. Radio access then opened the way to privacy threats, forcing passports to be secured with some access control.

The ICAO standard includes an optional Basic Access Control (BAC), based on 3DES [ISO10b], which essentially consists of making the reader prove that it knows a piece of information on the printed MRZ. This information called MRZinfo consists of the passport serial number, the date of birth of the person, and the expiration date of the passport. That is, BAC uses symmetric-key cryptography with an access key which is printed on the passport. Furthermore, MRZinfo has a low entropy (roughly 56 bits as explained in [ICAO06]). BAC is currently implemented in almost every passport, as the ICAO standard has been internationally imposed.

The BAC protocol is followed by some key agreement to open secure messaging. Again, it is all based on symmetric cryptography with a low-entropy initial key (the MRZinfo), so it does not resist passive adversaries.

The ICAO standard also includes an optional Active Authentication (AA) protocol which is based on a digital signature scheme. The MRTD authenticates itself by signing with its private key, a presumably random challenge from the reader. As this private key is securely stored and used in the chip of the MRTD, AA protects against cloning attacks but is time-consuming for the powerless chip. The AA protocol is currently implemented and used in Belgium and the Czech Republic. Unfortunately, AA is not secure against man-in-the-middle attacks [HR07] and leads to privacy concerns by adding the threat of *challenge semantics* [BSI09a]. A challenge semantics attack happens in the case where the reader chooses an unpredictable verifiable challenge, such as a signature of its location, date, and time. As the signature provided by the MRTD is transferable, it will attest to anyone trusting the reader that the MRTD was at a specific location, date, and time.

There are two clear advantages to the ICAO passports: the identities are unforgeable, and access to the chip requires knowing MRZinfo. Unfortunately, the drawbacks are many. First of all, the cryptographic protocols used do not resist passive adversaries. Since AA is seldom used, the ICAO standard does not resist cloning attacks. Furthermore, MRZinfo grants an unlimited permanent access: once the adversary obtains it, he can access the chip without the consent of the holder. Contrarily to popular belief, the release of DG2 and SOD is not privacy insensitive. Releasing DG2 means releasing an optimized picture which is used as a reference template for biometric recognition. Once an adversary obtains it, he can train himself to

match the template. Releasing DG2 can therefore ease identity theft. Furthermore, some countries, such as Switzerland, have put in place a national database storing all biometrics of their citizens. If the database gets compromised, identity theft will be even easier as the adversary will simply run a search on the closest match present in the database (excluding himself). Hence the assumption 2.3 in section IV of [ICAO06] is wrong.

> "The digitally stored image of the face is assumed not to be privacy-sensitive information. The face of the MRTD holder is also printed in the MRTD and can be readily perceived."

In addition, releasing SOD means providing transferable evidence of the correctness of the identity. For instance, it could be used as an undeniable identity proof against whistle blowers, which would compromise their safety if they need to remain anonymous.

6.5 EAC v1

The European EAC standard [BSI06, BSI08a] was made to add better protection for non-mandatory data groups such as DG3: the fingerprint template. It includes:

- secure messaging based on Elliptic Curve Diffie-Hellman [CR00];
- a chip authentication protocol, protecting against cloning attacks; and
- a terminal authentication protocol.

Terminal authentication is meant to be mandatory for accessing non-mandatory data groups, but mandatory data groups must remain readable without EAC due to the ICAO standard interoperability.

In the terminal authentication protocol, the reader proves that he owns the secret key associated to a given public key. Typically, this proof consists of signing a challenge from the passport. The public key has a certificate chain whose root belongs to the home country of the passport. That is, authorization is given to readers by signing a certificate with a given validity period. The problem with this method is that passports do not have any reliable clock. They keep a trusted past date in memory, which plays the role of a clock. When they check the validity of a certificate, they only check that the expiration date is posterior to the clock value. If verification succeeds and the issuing date of the certificate is posterior to the clock value, the clock value is updated. Clearly, passports which do not run terminal authentication often, will not even have a reliable approximation of the current date. Others may have a date which is accurate within a range of a few weeks. Consequently, a terminal certificate may be usable a long time after expiration.

The details of the general PKI required to authenticate readers will be given in the following section.

The advantage of EAC compared to ICAO, is the introduction of anti-cloning protection, a better key agreement resisting passive adversaries, and that readers are given time-limited privileges. One of the remaining problems is that revocation is based on a weak clock. Privacy issues relating to the release of DG2 and SOD to everyone remain. The hash of protected data groups also leaks from the SOD [BSI09a].

6.6 EACv2

EACv2 was initially released in 2008. The latest update was provided in the BSI TR-03110 Technical Guideline [BSI15a, BSI15b, BSI15c, BSI15d]. It was released in February 2015 as version 2.2. It specifies the mutual authentication between terminal readers and all kinds of MRTDs, including biometric passports.

The aim of EACv2, with its mutual authentication, is threefold. It first of all allows authorities to verify that a MRTD is genuine. It also allows authenticated terminals to access sensitive data contained in the MRTDs, such as fingerprints. Lastly, it provides a secure channel between the MRTD and the terminal. This authentication process relies on an international Public Key Infrastructure (PKI), described in [BSI09b] and in the EACv2 standard [BSI15c]. This PKI is mainly composed of three entity types : Country Verifying Certificate Authorities (CVCAs), Document Verifiers (DVs), and terminals. Each participating country will possess a national CVCA that will act as a national root authority. The national CVCA will be in charge of issuing national MRTDs and DVs certificates (especially foreign DVs certificates). DVs are organizational units within countries, in charge of managing a group of terminals, notably by issuing their certificates. Their role is to enable the certification link between its terminal readers and CVCAs. Hence they need to apply for a DV certificate at each CVCAs corresponding to the country of MRTD that might be encountered by its terminals. DVs are also in charge of creating and maintaining terminal certificates for each terminal location. The validity period and the access rights of the terminal certificate are inherited from the DV certificate. Obviously, these authorizations can be further reduced by a decision of the DV in charge of the terminal. In the same way, the validity period and the access rights contained in the DV certificate is decided by the CVCA issuing the certificate.

The access rights for all data groups are encoded in binary in each certificate, as an object identifier. These rights are set according to the role of the certificate holder (inspection systems, authentication terminals or signature terminals). A member in the certificate chain cannot provide more access rights than it has itself. Thus, to determine the access rights of a particular reader, the MRTD has to compute the boolean AND of all the binary authorizations contained in the certificate chain.

Furthermore, two types of terminals can be distinguished: integrated terminals and distributed terminals. An integrated terminal is a unique hardware device including a single reader. A distributed terminal is composed of a terminal control center, several readers, and a permanent, secure online channel between all readers and the terminal control center.

The EACv2 general authentication procedure is composed of four steps in the following order: Password Authenticated Connection Establishment (PACE), Terminal Authentication, Passive Authentication, and Chip Authentication. PACE is a Diffie-Hellman key agreement protocol based solely on a shared password. This password is either known by the MRTD bearer, or is directly printed on the MRTD. The goal of PACE is twofold: on one hand it provides a password based mutual authentication, and on the other hand it arranges a secure messaging channel with ephemeral symmetric sessions keys, one for encryption and another one for the Message Authentication Code (MAC). MACs are codes that help attest the authenticity of messages, even when sent over an insecure channel (see Section 2.2.3). Nevertheless, PACE yields a secure authenticated key agreement as proven by Bender, Fischlin, and Kügler in [BFK09]. Once PACE has succeeded, the MRTD is ensured that the terminal has knowledge of the shared password and thus gives access to its less-sensitive data. Moreover, all further communications are protected against eavesdroppers as secure messaging is put in place. However, an adversary with knowledge of the shared password, obtained either by guessing or by social engineering, will be able to mount a man-in-the-middle attack.

Terminal Authentication is then performed as the second step of the EACv2. Regarding terminal authentication and terminal revocation, no progress was made between version 2.01 in 2009 and the current version 2.20 of the EAC standard of 2015. Detailed explanations about them will be provided below. After the terminal has been authenticated, Passive Authentication enables terminals to confirm that a MRTD has not been altered. This step does not protect against cloning attacks. In order to achieve cloning protection, Chip Authentication is performed. This last step insures that the MRTD is genuine.

6.6.1 Terminal Authentication

A complete description of the terminal authentication can be found in Section 3.3 of [BSI15b]. It is essentially composed of three major phases. First, the terminal sends a certificate chain starting from the CVCA certificate corresponding to the MRTD country. The chain ends with the certificate of the terminal itself. In the second phase, the MRTD checks the certificates contained in the certificate chain with a Certificate Validation process (section 2.5 of [BSI15c]). The third phase consists of setting up an authenticated ephemeral Diffie-Hellman key pair for the terminal. The resulting ephemeral public key will then be used to secure messages from the MRTD to the terminal.

If the terminal authentication succeeds, the MRTD will grant access rights to its sensitive data, according to the *terminal effective authorization*. The terminal effective authorization is

derived from the certificate chain as the smallest authorization set present in all certificates of the certificate chain.

6.6.2 Terminal Revocation

The terminal revocation status is checked during the terminal Certificate Validation (section 2.5 of [BSI15c]). Surprisingly enough, the revocation process is performed only with the expiration date contained in the certificate and with a "Current Date" approximation stored in the MRTD. The major problem, as expressed in [CV09], is that MRTDs do not have a reliable clock. This is why they try to approximate the current date. Unfortunately, due to the requirements for this approximation, the "Current Date" could be outdated by as much as a month. Indeed, this update is executed solely with the date of certificate creation, contained in a certificate issued by the same country as the MRTD. Note that there is no passport control within the Schengen zone. For departures from the Schengen zone, an identity control will be required only at the last Schengen airport before a non-Schengen country. More information can be found in [Eur06]. As it is quite rare for a MRTD to encounter a terminal of its own country, the update will be executed with the date of certificate creation contained in foreign DV certificates. These are issued by the same country as the MRTD one.

Hence, a stolen terminal can still be used for a long period of time, even if its expiration date has passed. This is an important threat that must not be neglected. Without a proper terminal revocation scheme, a stolen terminal could be set up to use solely EACv1 without PACE, and thus be used to detect and target individuals or a specific group of persons, while the attacker is absent from the crime scene. Even in the case where EACv2 with PACE has to be used, if the shared password is compromised, then all sensitive data will be accessed after completion of the terminal authentication.

Oliver Bausinger from BSI claimed during the BIOSIG 2013 conference, that this issue is solved with distributed terminals and only integrated terminals remain vulnerable. Indeed, Section 1.2.1 of [BSI09b] mentions the following regarding integrated terminals:

> "The disadvantage of this architecture is, that a stolen reader can be used to perform Terminal Authentication at least as long as the current CV certificate is valid."

Moreover, Section 1.2.2.1 of [BSI09b] contains the same argument regarding distributed terminals, as that of Oliver Bausinger :

> "The advantage of this architecture is, that a stolen reader cannot be used for Terminal Authentication. Therefore each reader can be operated easily in an insecure environment."

Unfortunately the solution provided by distributed terminals introduces a single point of failure with the dedicated online terminal control center. As soon as this server fails, the entire terminal authentication procedure is stalled for all readers. The permanent online channel required for them is also a potential target for attackers. In highly visited border controls, this can become a major drawback or even a potential threat. An attacker only needs to jam communication between readers and the terminal control center in order to paralyze an entire border control.

6.7 Conclusion

Putting aside the weak ICAO standard, EACv2 resolves one of the issue of EACv1, namely the privacy issue linked to releasing DG1, DG2, and SOD. The main difference introduced by this version is in the order of authentication between a chip and the terminal that is attempting to read it. In this new specification, the terminal authentication must be performed before the chip authentication. EACv2 even introduces a replacement for BAC, named PACE. PACE is a state-of-the-art password-based access control resisting active attacks. Another improvement is that the access password for PACE is now a specific secret printed inside the passport and no longer private data which has other purposes such as the MRZinfo.

This modification could be considered, at first glance, a major improvement. Indeed, by forcing authentication of the terminal before the chip authentication, the release of DG2 and SOD is restricted to officially allowed terminals only. However, this is not the case in the full view of the specifications. A careful read of the specifications of the EACv2 in [BSI15a] reveals the following in a footnote of section 2.4.1:

> "For an ICAO-compliant ePassport application the MRTD chip MUST grant access to all less-sensitive data (e.g. DG1, DG2, DG15, etc. and the Document Security Object)."

What this note states is that if compatibility with ICAO is required, then the MRTD must behave as in the ICAO standard. In other words, any fake terminal reader can require the MRTD to use the crippled ICAO standard.

Furthermore, the date contained in the MRTD is still an approximation of the current date. The date is updated only with national domestic certified dates, by means of *certificate effective dates* (date of the certificate generation), contained in a national domestic CVCA certificate, a DV authorization certificate issued by the national domestic CVCA, or an *accurate terminal* certificate. The latter is a terminal certificate issued by an official domestic DV. As a MRTD will rarely encounter a domestic terminal, it is more likely that its date will be updated through the certificate effective date contained in a foreign DV. Hence the revocation of terminals is far from being solved with the current EACv2 standard.

Chapter 7

Enhancing the EAC

This chapter, which is based on a revised version of [CV09], and on an extension of [Cha13], proposes several enhancements and solutions for the next EAC upgrades. Section 7.2 emphasizes the prior attempts to solve the issues associated with EACv2. Section 7.3 proposes a light hardware modification for new passports. Section 7.4 discusses an ICAO standard improvement that consists of replacing its Basic Access Control (BAC) with the Password Authenticated Connection Establishment (PACE) of EACv2. In the event that no hardware modification will be tolerated, Section 7.5 suggests to increase domestic controls in order to improve the time-based revocation of terminals. Furthermore, Section 7.6 presents a full solution for terminal revocation, which limits itself to a software upgrade.

7.1 Introduction

Two types of threat are studied in this chapter. The first one is related to the threat of a stolen integrated terminal device. These are considered to be Portable Computing Devices (PCD) in the Technical Guideline TR-03110 [BSI15a, BSI15b, BSI15c, BSI15d]. An integrated terminal, as explained in [BSI09b], consists of a single reader with an integrated hardware security module and a proximity coupling device. Moreover, a stolen integrated terminal could still be used to read MRTDs, as long as its certificate has not expired. This threat applies even with an expired certificate if the date approximated in the MRTD is outdated. Hence there is no real revocation system present for terminals. This is a known problem for the BSI, and is even mentioned in [BSI09b] (Section 1.2.1). The second type of threat originates from an inside attack. This incites for the study of the threat case where a compromised terminal has remained in place, acting maliciously. With the current standard, a stolen or compromised terminal could be used to target a group of persons, for instance by nationality, or a specific person, such as Politically Exposed Persons (PEPs).

The implications of these threats are threefolds. First of all, they introduce an obvious privacy breach in the sense that any compromised integrated terminal will have an illegitimate access

to all MRTD data including biometrics. From there, an attacker can filter and target specific individuals, or even groups with specific attributes, such as a specific nationality. Moreover, the terminal can be used to acquire all information from all MRTDs that come in geographic proximity with it, in order to build an illegitimate database of biometrics. With this kind of database, attackers can train themselves and select the closest match for a cloned identity.

Lastly, efficiency needs to be taken into account. In [Fri], it is mentioned that more than 56 millions passengers traveled through Frankfurt airport in 2011. As around half of them are only transfer passengers, and thus do not necessarily need a passport control, big hubs need to process more than 2 million passport checks per month.

7.2 Prior and Related Work

As shown in the previous chapter, there is still room for improvement in the EAC standard. In that regard, some results have already been proposed.

In 2009, Monnerat, Pasini, and Vaudenay [MPV09] constructed an Offline Non-Transferable Authentication Protocol to achieve a Zero-Knowledge proof of knowledge of a valid SOD, which has been neglected by the BSI.

Regarding the issue of terminal revocation, it has received only a small amount of interest as the BSI community is convinced that the Password Authenticated Connection Establishment (PACE) protocol mitigates this threat, as explained in [BDFK12]. Indeed, when executing EACv2, PACE is the initial phase before Terminal Authentication. After its successful completion, the MRTD is ensured that the terminal has knowledge of a shared password, and can proceed with Terminal Authentication. However, no guarantees are provided in the obtention of this password. If the shared password has been obtained by social engineering, or read directly by eavesdropping on the MRTD, then a successful terminal authentication will allow the stolen terminal to access all sensitive data contained in the MRTD. This issue has been raised by Belguechi et al. in [BLR12]. Unfortunately, the BSI concentrate on the protection of biometric data and do not provide a solution for terminal revocation.
Li et al. in [LZJX10] also mention the threat of terminal revocation, but concentrate on presenting the Singapore solution that implicates Authorized Smartcard with Identity Based Cryptography. Hence, to solve terminal revocation they require heavy hardware modifications.

In [BB13], Buchmann and Baier presented two solutions for terminal revocation. Both of their solutions imply that MRTDs communicate securely with a trusted home server. In their first solution, this communication is needed twice: it is first used to retrieve the current authenticated and precise date with the Network Time Protocol (NTP), and it is also used to identify the terminal revocation status by accessing an Online Certificate Status Protocol (OCSP) server. In their second solution, MRTDs transfer the entire authentication check to the trusted home server with the Server-based Certificate Validation Protocol (SCVP).

Not only do both of these solutions require new heavy hardware incorporation, they also require the establishment of a secure and permanent high speed bandwidth connection between dedicated country servers and every potential terminal in the world. This might become an issue for mobile terminals aboard international cruise ships, with poor and unreliable connectivity. Furthermore, both of their solutions introduce several *single points of failure*. Indeed, if one of the network links or if one of the servers for either NTP, OCSP, or SCVP fails, both of their entire terminal revocation solutions become unusable.

7.3 Light Hardware Improvement

Currently, it is easy to distinguish between passports from different countries without any direct contact with them. The only way to protect against this is to prevent the chip from responding. In order to avoid traceability of passports, the solution that people currently have is to place their MRTD in a Faraday cage. Obviously this solution is cumbersome. For the case of biometric passports, a better solution would be to incorporate an *RFID switch* to deactivate the chip. Some sensors could even detect if the passport is opened or closed and manipulate the switch accordingly. This last solution can be accomplished by placing a secondary RFID tag antenna in the back cover of the passport, and joining both antennas with a NAND gate. When the passport is closed, the RFID tag would simply ignore all discovery signals sent by readers, as both antenna will provide a power source. In order to interact with the RFID tag, the passport would need to be opened, allowing for a single antenna to be powered. This solution is logical, as the access password for PACE printed inside the passport is supposed to be scanned by border patrols. The main drawback, although being cheap, is that it involves a small physical modification to passports (for instance an additional RFID tag antenna and a NAND gate).

7.4 Improving ICAO Standard

Several changes need to be made to the current EACv2 specifications as well as to the ICAO standard. The first issue to be considered for the ICAO is that *BAC should be abolished* and replaced by PACE. For interoperability between the EAC and the ICAO standard, the latter should stop mandating the availability of DG1, DG2, and SOD without PACE from the EAC. Moreover, EAC would have to be implemented outside Europe in order to fully deploy its capacity. As for the EAC and ICAO standards, they only require eliminating a few lines in their specifications. This proposed enhancement has been taken into account, and the ICAO working group ISO/IEC JTC1 SC17 WG3 mentioned in [ICAO13] that:

> "At present the fact that BAC MUST always be present on the eMRTD ensures that inspection systems that do not support PACE (yet) will still be able to access the

> MRTD's chip. To access eMRTDs supporting only PACE, inspection systems MUST support PACE. In its meeting on 19-21 February 2013 the NTWG concluded that as of the date 01 January 2018 eMRTDs supporting only PACE will be considered to be ICAO compliant. The chosen date should provide enough time for inspection system owners and vendors to implement the necessary modifications to their systems."

Deployment does not necessarily imply a heavy PKI for terminals. A country not ready to have such a PKI could still use a dummy one with a single key shared between all readers. The passport issuing country, aware of this, could adjust the read access to mandatory data groups and maintain the possibility of stopping the renewal of a certificate for this key if the reading country does not make enough effort to avoid leakage of its secret key. EAC-reading is a matter of software update and is inexpensive. A first step has been made by the ICAO towards mandatory pure EAC, however BAC will still be present for MRTDs supporting it.

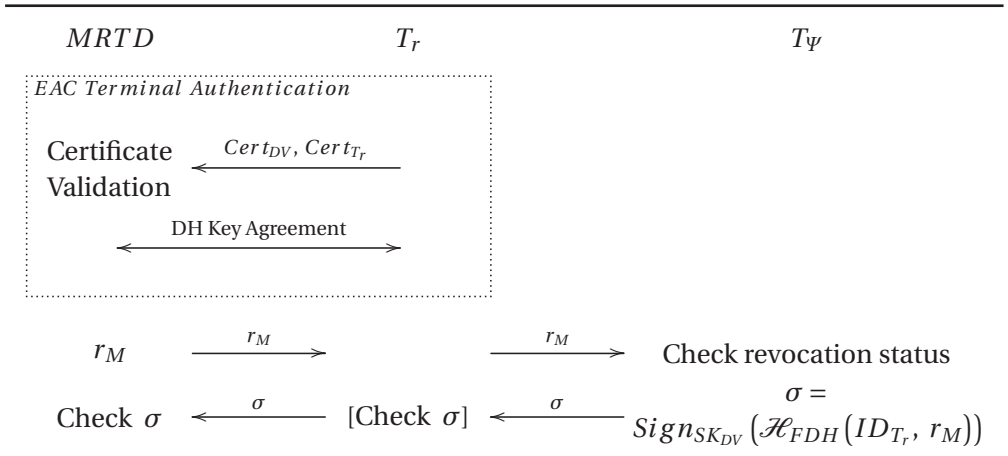
7.5 Improving Behavioral Practices

To be more accurate in the date contained in the MRTD, a solution would be to have identity checks even when leaving a domestic country or a community space if the community space members trust each others. For instance, some domestic clock-update booths could be made available on a voluntary basis before departure. As the identity check will correspond to an interaction with an *accurate terminal*, the date in the MRTD will be updated with the terminal *certificate effective date*. The date contained in the MRTD is still an approximation in this scenario, but with a reduced date error when compared to EACv2. Ideally, future chips should be equipped with a real clock. If no improvements are made to the EAC standard, the only solution left for holders of MRTDs to maintain their privacy, is to shield their MRTD in a Faraday cage.

7.6 Solving Terminal Revocation

The new method presented in this section uses threshold signatures in order to verify the revocation status of terminals. The background is explained in detail in Section 2.5 and in particular in Sections 2.5.2 and 2.5.3. Document Verifiers (DVs) in the EAC standard are here assumed to be trusted participants. In general, several terminals are present. If the number of terminals is considered too low, this scheme can easily be modified to provide equivalent properties. It is also assumed that a communication channel between terminals exists. This is a common feature of terminals nowadays, such as the 3M^{TM} Mobile ID Reader. Moreover, the modifications needed to enable this method are solely software upgrades: no hardware modification to MRTDs is required.

Let us now see how to go from threshold signatures to terminal revocation. The main idea is to introduce terminal collaboration in order to achieve terminal authentication. Terminal revocation will thus be achieved with the help of neighboring terminals. Protocol 7.1 depicts the general view of how the EAC terminal authentication should be augmented to provide a better revocation mechanism. The terminal interacting directly with the MRTD will be called the requesting terminal T_r, as it will request collaboration from neighboring terminals to achieve authentication. The set of terminals participating in a specific terminal authentication is denoted by T_Ψ. This set includes T_r. Furthermore, Ψ will be the index set of these terminals.



Protocol 7.1 – Augmented terminal authentication

The additional interactions needed for terminal revocation are added after the EAC terminal authentication is performed and before giving access rights to T_r for the MRTD sensitive data. They consist of three main steps. In the first step, the MRTD generates a fresh random nonce r_M, that will be transmitted to T_r and forwarded to the set T_Ψ. In the second step, T_Ψ will check the revocation status of T_r. As terminals have real clocks and better computation capabilities than MRTDs, they will be able to check this revocation status much more efficiently. In the third and final step, T_Ψ will produce, with the shared DV secret key SK_{DV}, a full domain hash (FDH) threshold signature σ. This signature will be performed on the MRTD challenge r_M joined with the identity ID_{T_r} of the requesting terminal. This signature σ will then be forwarded to the MRTD which will check it against the DV public key. If the check succeeds, then the MRTD will be ensured that the terminal T_r is authentic and non-revoked.

Following the classification of authentication protocols proposed by Park, Boyd, and Dawson in [PBD00], this proposed addition is an *origin authentication* protocol with *forced challenge*, where the prover signs with his secret key a random nonce generated by the verifier. The inclusion of the terminal T_r identity in the signature is necessary to avoid a Lowe attack [Low96].

The identity of the MRTD is unnecessary when sending the challenge as the objective is only to provide a strong argument that T_r is not revoked. Moreover, linkage to this identity might raise a privacy issue. The disclosure of the signature would reveal the location of the MRTD at a given time if it is joined with a timestamped signature.

The security assumptions will be specified first. Detailed explanations will then be provided on how to extend terminal authentication in order to achieve a better terminal revocation. Thereafter, some efficiency improvements will be discussed. The security requirements will then be explained, a complete security proof will be provided, and a general analysis will conclude this section.

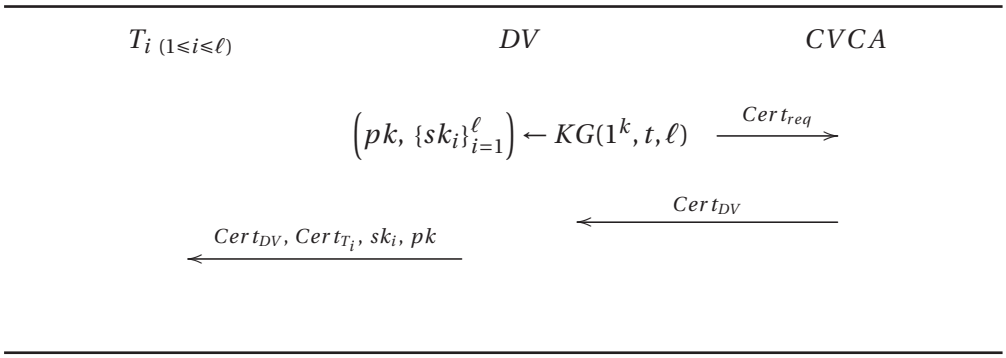
Security Assumptions

Regarding the environment, the same structure of participants as the EAC model is assumed, however some clarifications are provided. Each DV is responsible for ℓ terminals (ℓ differs from one DV to the other), where DVs and terminals are polynomial-time algorithms. DVs play the role of trusted authority amongst their terminals. The existence of secure and authenticated channels between all ℓ terminals is assumed. This is easily achieved with public key encryption as it is the same DV (a trusted party) that issued every terminal key pairs. When a terminal is stolen, its certificate will be revoked. This revocation will disable its use. Moreover, the lack of online connectivity should apply only to CVCAs and DVs as they are Public Key Generators. As such, they should be turned offline once their keys setup generation has been achieved (as explained in [Sha84]). This is not the case for terminals. As for MRTDs, recall that they have no internal clock. Regarding time, MRTDs should therefore consider it as indicative but not decisive.
Furthermore, attackers are assumed to be *computationally bounded*. Focus will be placed on threats relating to terminals, as they are somehow neglected in the current EAC. Nevertheless, both CVCAs and DVs are assumed to be honest. The threshold security requirement, where the adversary can corrupt up to t terminals among $\ell \geqslant 2t+1$, is also assumed to hold. This last assumption can be lessened if proactive security is included (Section 2.5.2). After each revocation, the value '0' would be shared amongst the remaining valid terminals, and added to their current shares. More details on the matter will be explained in the general analysis of Section 7.6. With proactive security, t has to be set such that no more than t terminals could be corrupted before a resharing.
Adversaries will be expected to be either *passive adversaries*, where attackers corrupt targets by reading their contents, their secrets, and all the communications involving them, or *active adversaries*, where attackers will additionally be allowed to change the behavior of corrupted terminals. Lastly, adversaries will be restricted to *static adversaries*, meaning that the adversary will select which terminals to corrupt before the start of the protocol. Moreover, the adversary is free to corrupt them whenever he wants to. When a terminal is corrupted, all his communications will be revealed to the adversary. *Dynamic adversaries* are set aside, as the corresponding solutions will induce a high loss of efficiency. Nevertheless, it would still be

possible to handle dynamic adversaries by using a threshold signature scheme secure against them [LY12].
Regarding revocation, only the case of *accurate revocation* is considered, meaning that corrupted terminals are immediately identified and thus revoked, and non-revoked terminals are considered honest and non-corrupted. This limitation makes sense as the initial goal of the suggested solution is to protect MRTDs against stolen terminals. Corrupted and non-revoked terminals are outside the scope of the adversarial model considered. It is important to note here that any non-revoked malicious participant will always be able to succeed in an authentication if the participant behaves honestly. Moreover, the threat of a cloned terminal is not covered with the following solution, as it will be explained in more detail in the general analysis of Section 7.6.

Augmented Terminal Authentication

Protocol 7.3 gives the general structure of the additional part to the current terminal authentication protocol. The required Setup phase, depicted in Protocol 7.2, is very similar to the original EAC one.



Protocol 7.2 – Terminal authentication setup

DVs are in charge of the setup phase. They will first run a key generation algorithm KG to obtain the system public key pk and the different terminal secret key shares sk_i intended for each terminal T_i. The secret key shares sk_i are shares of the DV secret key SK_{DV}. They are computed such that every terminal authentication will require the collaboration of at least $t+1$ terminals. Hence, the following scheme tolerates up to t corrupted terminals. As long as $t+1$ honest terminals are available, terminal authentication will be able to proceed. Recall that Ψ is the index set of terminals participating in a specific terminal authentication. Hence $\Psi \subset \{1, \cdots, \ell\}$ and $|\Psi| \geqslant t+1$. Furthermore, the system public key pk includes the DV public key PK_{DV}, the verification key vk_i of each terminal, as well as some system parameters.

After KG has been run, the DV will have to contact CVCAs from every other country with a certificate request, in order to obtain its DV certificates. Hence, a DV will have one certificate for each country whose MRTDs may likely encounter the terminals of that DV. This is already the case with the current EAC. The difference in certificates is that they will additionally include PK_{DV}. They will not include the entire pk as only PK_{DV} is used in the interaction between MRTDs and terminals. Moreover, certificates will contain additional information regarding how many terminals are required to collaborate with the requesting terminal T_r in order to complete its authentication (parameter t), as well as how many terminals are present under the DV (parameter ℓ).

The DV first receives its certificates. Each terminal i then receives the public key of the system pk, its corresponding secret key sk_i, its certificate and the DV certificates, all from the DV. Once the Setup phase has been completed, SK_{DV} can be safely erased. Future interactions will include terminals and MRTDs only. Hence the DV can be turned offline as described in the EAC standard.

$MRTD\ (PK_{DV})$		$T_r\ (pk,\ sk_r)$		$T_{i\neq r}\ (pk,\ sk_i)$
$M \in_R \mathcal{M}$	$\xrightarrow{M}$	$\widetilde{M} = \mathcal{F}(ID_{T_r},\ M)$	$\xrightarrow{M}$	Check revocation status
		$\sigma_r = \Sigma_{i=r}(\widetilde{M})$		$\widetilde{M} = \mathcal{F}(ID_{T_r},\ M)$
		Check $\Sigma_v(\sigma_i, \pi_i)$	$\xleftarrow{\sigma_i, \pi_i}$	$(\sigma_i, \pi_i) = \Sigma_i(\widetilde{M})$
Check $V_\sigma(\mathcal{F}(ID_{T_r},\ M))$	$\xleftarrow{\sigma}$	$\sigma = \Sigma_c(\{\sigma_i\}_\Psi)$		

Protocol 7.3 – Terminal authentication with revocation

The terminal authentication takes place after the setup phase and the Certificate Chain Validation process. Protocol 7.3, which depicts the general terminal authentication, makes use of Section 2.5.2 notations. At first, a MRTD will select a random challenge M in the message space $\mathcal{M}$. It will then challenge the requesting terminal T_r with this random challenge M. Moreover M must be independent from the MRTD identity, otherwise a tracking threat would arise in relation to location privacy. Indeed, in the case where M is related to the MRTD identity, the signature will prove that a given identity was at a specific location at a specific time. Maliciously replaying this challenge does not pose any real threat as it will only reveal a previous valid signature for the authenticity and the non revocation status of a given terminal. As the communication channel between the MRTD and the terminal is only secured with PACE, there is no authentication guarantee or revocation status check provided on the terminal

certificate. They will be provided on the ability of the terminal to perform the signature on the MRTD challenge. After the certificate is validated by verifying the correctness of the signature, the MRTD will be able to use the keys contained in the certificate to establish a secure and authenticated channel.

In order to get a valid signature on the MRTD challenge, the requesting terminal T_r will have to collaborate with at least t other terminals. The *revocation* process takes place during the terminal collaboration. It will be the role of other terminals to determine the revocation status of T_r. To do so, an honest T_r will contact solely non revoked terminals for the collaboration. Any standard strong revocation mechanism can then be used here. The basic solution is to apply Certificate Validation as described in Section 2.5 of [BSI15c], except that a real clock can now be used. The advantage of this solution is that no additional hardware is introduced. Nevertheless, more complex solutions can also be used, such as Certificate Revocation Lists (CRL) or the Online Certificate Status Protocol (OCSP), if an OCSP responder is set up under the DV authority and is just for terminal interactions.

The constant participation of the DV in the revocation process should be avoided as it breaks the principle of closing the Public Key Generator (PKG) after key generation (as mentioned in [Sha84]). Furthermore, the case of OCSP introduces two drawbacks. It requires the introduction of additional hardware, and the OCSP responder becomes a single point of failure. The CRL solution should be favored, as it can be manually pushed towards terminals when necessary. The CRL size remains small as it targets only collaborating terminals under a same DV.

If a CVCA considers that the threshold t used in an organizational unit managed by a DV is too low, it can request the participation of a special terminal that will act as a revocation server. Nevertheless, as was the case for the OCSP, this method introduces a single point of failure with the revocation server. Ideally, the set up of the organizational unit under a DV should include enough terminals for the revocation process. As CVCAs provide foreign DVs the ability to read their passport, it would be desirable that these DVs protect this privilege, and avoid its misuse. If the number of terminals required is low (in a hotel, for example), then these terminals should join the infrastructure of another existing DV organizational unit.

If the requesting terminal T_r is revoked, then its request will simply be ignored by other honest terminals. If the T_r status is still valid (not revoked) then partial signatures σ_i can be computed and sent to it, possibly with verification proofs π_i. Moreover, the partial signatures will also contain the identity of the terminal T_r requesting this signature. To include this identity, terminals will use a full domain hash function $\mathscr{F}$ on the MRTD challenge M and on the T_r identity ID_{T_r}. The hash outputs should cover the full input domain of the signature scheme, in other words the message space of the signature scheme. T_r will then collect all valid t partial signatures and combine them with its own to create a full domain hash (FDH) threshold signature σ on the MRTD challenge. This global signature σ will be sent to the MRTD as a proof of authenticity and its non revocation status.

It is important to note here that $\mathscr{F}$ should be second-preimage resistant. This requirement is a necessary condition to avoid relay attacks. For a given legitimate challenge M and an honest terminal ID_{T_r}, an attacker with the ability to undertake a second-preimage attack could forge a request $M_{\mathscr{A}}$ to T_r such that $\mathscr{F}(ID_{T_r},\ M_{\mathscr{A}}) = \mathscr{F}(ID_{T_{\mathscr{A}}},\ M)$, where $T_{\mathscr{A}}$ is a terminal corrupted and controlled by the adversary. The requirement that all terminals compute independently $\mathscr{F}(ID_{T_r},\ M)$ is also essential. If this computation is achieved solely by the MRTD and transmitted to T_r for signature, an adversary controlling a rogue terminal could mount a man-in-the-middle attack. Such an attack would consist of three main steps. First, the adversary would initiate an interaction with a MRTD by sending it his identity $ID_{T_{\mathscr{A}}}$. Then the adversary, pretending to be a MRTD, would initiate an interaction with a legitimate terminal T_r. In this interaction, the adversary would simply ignore the identity of T_r. To obtain a valid signature on $\mathscr{F}(ID_{T_{\mathscr{A}}},\ M)$, the adversary can simply forward it to T_r, as $\mathscr{F}(ID_{T_{\mathscr{A}}},\ M)$ has been computed and sent by the MRTD. If $\mathscr{F}(ID_{T_r},\ M)$ is computed solely by T_r and later by the MRTD in the verification step, terminals T_i would have no means to verify if the terminal interacting directly with the MRTD is indeed T_r, and another similar attack could be achieved.

Once the MRTD receives the global threshold signature σ, the MRTD will have to verify it with the global public key of the DV, PK_{DV}. If the check is successful, the MRTD can be ensured that either the terminal is able to forge signatures on behalf of the DV (for instance the terminal knows the DV secret or it can cheat on its revocation status), or that the terminal has gone through a threshold signature involving revocation checks. As the DV is assumed to have correctly achieved the initial setup and that terminals communicate over an authenticated channel, the MRTD is ensured of the non revocation status of the terminal.

At this point, any efficient, unforgeable, robust, and secure threshold signature scheme can be used. Optionally, it can also be proactive secure. All of these properties are achieved by the threshold RSA signature of Shoup [Sho00], explained in Section 2.5.3. This choice is favored for its efficiency and simplicity. A detailed description of the *KG* in the DV setup phase follows in Protocol 7.4, whereas the description of the protocol is given in Protocol 7.5. Both cases follow the notation introduced in Section 2.5.3.

In the *KG*, note that some precomputations are done by the DV for its terminals, namely the Δ computation, as well as the computation of parameters a and b. Furthermore, if proactive security is targeted, then the *KG* must also provide $\tilde{n}$ as part of the DV secret key SK_{DV}. The latter will be explained in the general analysis of Section 7.6. In all cases, once the setup phase has been entirely completed, the DV secret d from SK_{DV} can be safely erased. Note, however, that this is not meaningful in the case of proactive security because p and q can be recovered from $(n,\ \tilde{n})$, and hence computing the inverse of $e \pmod{\tilde{n}}$ is easy.

After the setup phase, the threshold signature of Shoup [Sho00] is used and applied to the general case (Protocol 7.3), which results in Protocol 7.5. The Lagrange coefficients $\lambda^{\Psi}_{0,\ i}$ are computed as in equation 2.3. For obvious reasons, the best choice for $|\Psi|$ is $|\Psi| = t+1$. This implies that T_r will only need to contact t other non revoked terminals. Moreover, after

receiving their credentials, each terminal can define a preferred set Ψ, which would include themselves. This would allow them to precompute the Lagrange coefficients corresponding to their set Ψ. Nevertheless, if a terminal from their preferred set Ψ fails to reply or fails to provide a valid verification proof, then the requesting terminal can contact other terminals and recompute the Lagrange coefficients for the new set Ψ. As for the choice of the one-way permutation $\mathscr{F}_{T_r}(ID_{T_r},\ M)$, it can only be based on SHA-512 as this is the best hash function implemented in current MRTDs. In order to provide a uniform output in $\mathbb{Z}_n^*$, which is the message space of the chosen signature scheme (threshold RSA), Bellare and Rogaway suggested in [BR96] to concatenate the same hash function where the hash input would be appended with a constant and a counter ctr_i. As ID_{T_r} can be used as the constant, $\mathscr{F}$ would correspond to:

$$\mathscr{F}_{T_r}(ID_{T_r},\ M) = \text{SHA-512}_0(ID_{T_r} \| ctr_0 \| M) ... \text{SHA-512}_i(ID_{T_r} \| ctr_i \| M) ...$$

Furthermore, to obtain the right output length, the last hash block can be truncated as suggested by Bellare and Rogaway in [BR93]. Unfortunately, the output will not be perfectly uniform in $\mathbb{Z}_n^*$, as a modular operation will still be required. Last but not least, there is a $\frac{(n-\varphi(n))}{n}$ probability that $\mathscr{F}_{T_r}$ fails to provide a valid output, corresponding to the cases where the output falls in $\mathbb{Z}_n \setminus \mathbb{Z}_n^*$.
An efficiency gain can be obtained by altering the way in which the verification proofs are conducted. This will be explained in the next section.

DV

Choose $\boldsymbol{n}$ as a composite RSA modulus from 2 safe primes:
$\boldsymbol{n} = (2p+1)(2q+1)$; and $\tilde{n} = pq$.

Pick $\boldsymbol{e}, \boldsymbol{d}$ such that $\boldsymbol{ed} = 1 \pmod{\tilde{n}}$.

Set $\boldsymbol{SK_{DV}} = \boldsymbol{d}$ and $\boldsymbol{PK_{DV}} = (\boldsymbol{n}, \boldsymbol{e})$.

Set $\boldsymbol{\Delta} = \ell!$.

Pick $\boldsymbol{v} \in_R QR_n$, where QR_n is the subgroup of squares in $\mathbb{Z}_n^*$.

Compute $\boldsymbol{a}, \boldsymbol{b}$ such that $\boldsymbol{a} \cdot 4\Delta^2 + \mathbf{b} \cdot e \equiv 1 \pmod{\tilde{n}}$
(with the Extended Euclidean Algorithm).

Set $\boldsymbol{\Omega} = \{0, \cdots, 2^{\|n\|+2L} - 1\}$, where L is a security parameter (at least 128 according to [Sho00]).

Set $\mathscr{H}$ as a hash function that has L bits output.

Pick $\boldsymbol{ID_{T_i}} \in_R \mathbb{Z}_{\tilde{n}}^*$, for all $1 \leqslant i \leqslant \ell$.

Pick a function $f(x) = \sum_{i=0}^{t} f_i x^i \pmod{\tilde{n}}$

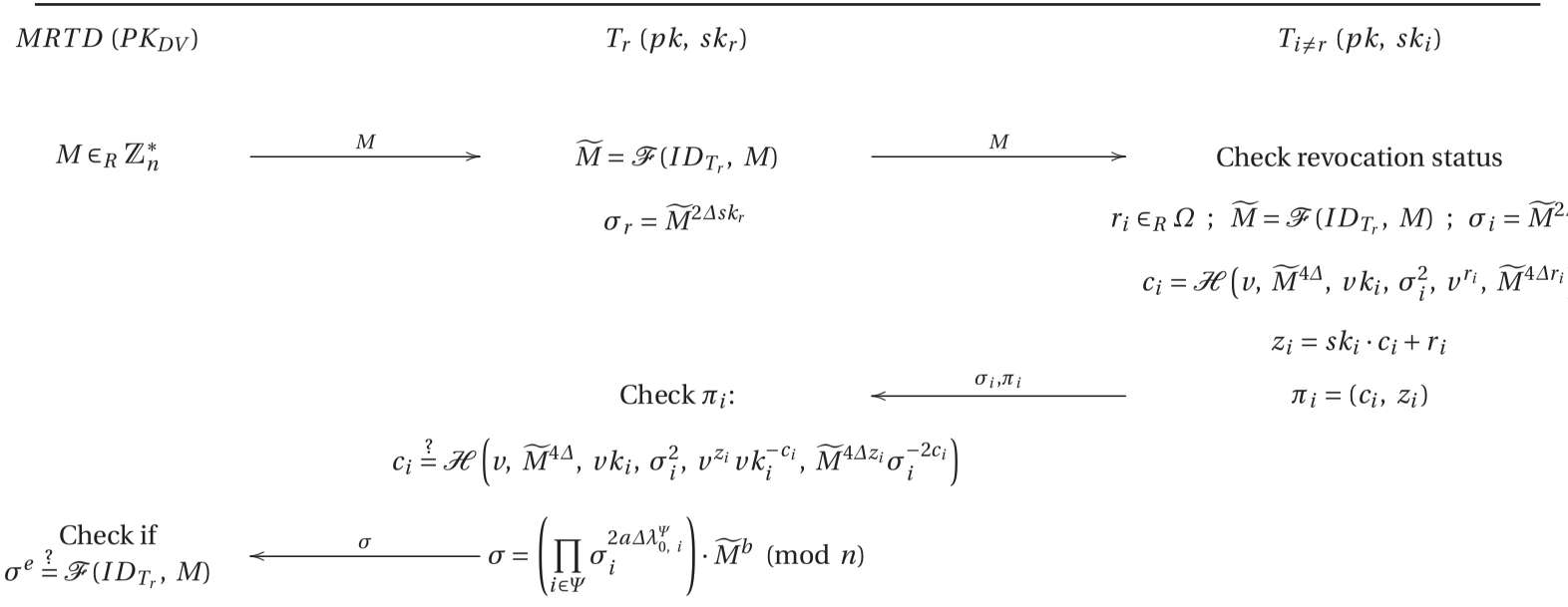
with $f_0 = SK_{DV}$ and $f_{i \neq 0} \in_R \mathbb{Z}_{\tilde{n}}$.

Set $\boldsymbol{vk_i} = v^{sk_i}$, where $sk_i = f(i)$ for all $1 \leqslant i \leqslant \ell$.

Set $\boldsymbol{pk} = \left(PK_{DV}, \Delta, v, a, b, \Omega, \mathscr{H}, \left\{ID_{T_i}, vk_i\right\}_{\forall i}\right)$.

$$DV \xrightarrow{\boldsymbol{Cert_{req}}} CVCA$$
$$DV \xleftarrow{\boldsymbol{Cert_{DV}}} CVCA$$
$$DV \xrightarrow{\boldsymbol{Cert_{DV}}, \boldsymbol{Cert_{T_i}}, \boldsymbol{sk_i}, \boldsymbol{pk}} T_i$$

Protocol 7.4 – DV key generation and setup



Protocol 7.5 – Terminal authentication with revocation

Efficiency Analysis and Enhancement

In Protocol 7.5, the MRTD computation is dominated by one single exponentiation. The terminal communicating directly with the MRTD and in charge of combining the partial signatures, has a computational complexity dominated by $(5t+4)$ exponentiations. However, this computational cost can be reduced to $(t+4)$ exponentiations as explained below. For the collaborating terminals, the computational cost is dominated by 4 exponentiations. Simple squaring is considered as a multiplication.

With regard to computational costs, several modifications can be made to reduce them. First, the terminal in charge of combining partial signatures could perform the robustness checks only if the resulting combined signature is invalid. Hence instead of computing $4t+1$ exponentiations, the validity of the signature can be checked first with a single exponentiation, as depicted in Protocol 7.6.

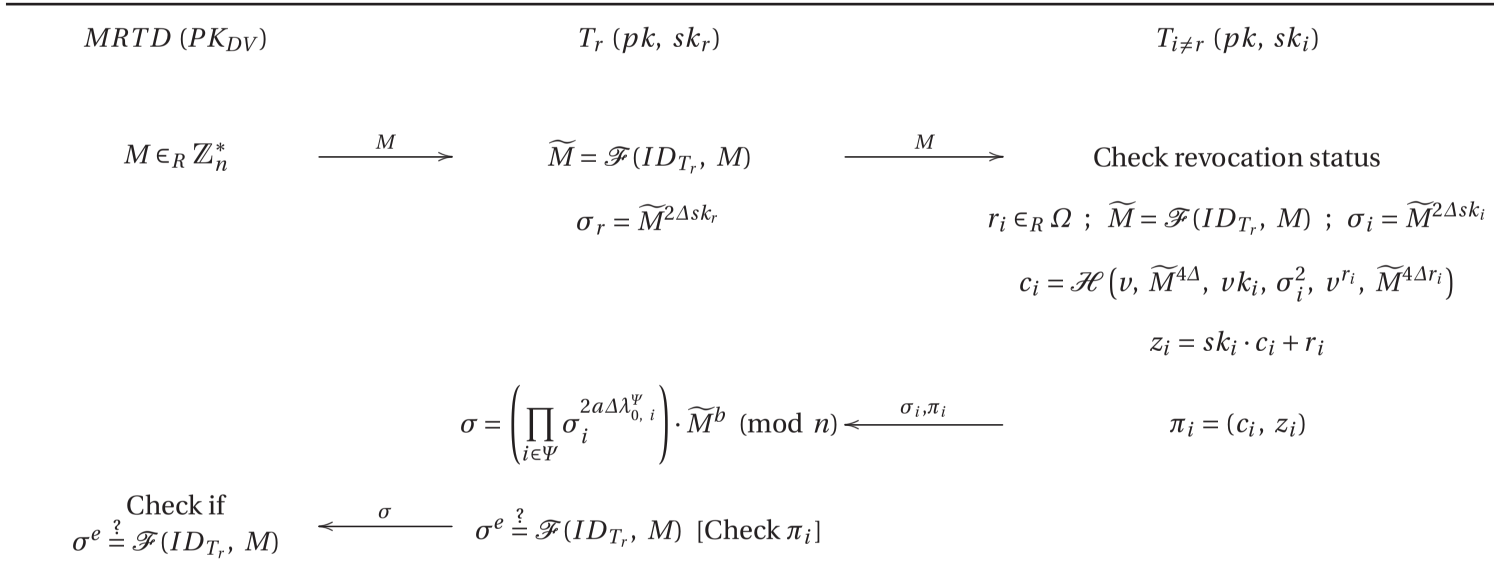
The overhead, in time, should be less than 0.1 seconds, assuming 30 MHz CPU for MRTDs, 520 MHz CPU for terminals, 802.11g wireless communication between terminals (net average of 22 Mbit/s) and 200 Kbit/s communication speed between MRTDs and terminals. Each message sent is around 1 Kbit except the messages from collaborating terminals that are around 3 Kbits.

Note that in Protocol 7.4, the DV precomputes Δ and $(a,\ b)$. It can go even further by precomputing the Lagrange coefficients $\lambda_{0,j}^{\Psi}, \forall \Psi$, and storing them in each terminal during the set up phase. The drawback of this method is that it requires a storage space in terminals. This can be an issue as there are $(t+1)C_{\ell}^{t+1} = \frac{\ell!}{t!(\ell-t-1)!}$ Lagrange coefficients to compute.

Furthermore, in the case of a large ℓ ($\ell > 100$), the exponentiation by Δ to obtain the general signature will greatly slow down the system. In this scenario, the threshold signature scheme of Gennaro et al. [GHKR08] will be preferable as it would be more efficient.

The existence of *multisignatures* should also be noted. These are a type of threshold signature, where the identity of signers is provided in the general signature. However, even the latest result in multisignatures that can be used here, namely the scheme from Boldyreva [Bol03], would imply a significant decrease in efficiency.

Finally, a modest efficiency gain could be obtained by using the threshold signatures of King [Kin00], which are derived from the Desmedt-Frankel [DF94] scheme. However, the gain in efficiency achieved necessitates a more complex implementation and a higher storage requirement.



Protocol 7.6 – Efficient terminal authentication with revocation

Security Requirements

The solution suggested here for terminal revocation is simple but uncommon. Authentication is achieved by means of authorization to perform a threshold signature given after a revocation check. Hence, the security requirements have to be clarified in order to provide an acceptable security proof. They include the following:

- *MRTDs privacy*;
- signature scheme *unforgeability*, *threshold security*, and *robustness*;
- *randomness freshness*;
- *signature freshness* and second preimage resistance for the hash function;
- *revocation implications*; and
- *completeness* and *soundness*.

Furthermore, the scheme can have *proactive security* as an optional requirement, depending on the complexity tolerated and the security targeted.

Before going into the details of these requirements, three oracles that will be accessible by the adversary for the security proof are defined: $\mathcal{O}_{\Sigma_i}$, $\mathcal{O}_\sigma$, and $\mathcal{O}_p$. Passive adversaries will only be allowed to query $\mathcal{O}_p$, while active adversaries may use all three of them.

Oracle $\mathcal{O}_{\Sigma_i}(ID_{T_r},\ M,\ \{ID_{T_i}\})$. Upon being queried with a challenge M, an identity ID_{T_r}, and a set of identities ID_{T_i}, the oracle $\mathcal{O}_{\Sigma_i}$ checks the revocation status of all identities. If ID_{T_r} is revoked, then $\perp_i$ is returned for all ID_{T_i}. Otherwise, $\mathcal{O}_{\Sigma_i}$ runs the partial signature algorithm of the threshold signature scheme for each non-revoked identity ID_{T_i} with input message $\mathcal{F}\left(ID_{T_r},\ M\right)$. At the end, the oracle returns all partial signatures σ_i with their verification proof π_i, corresponding to the non-revoked identities in $\{ID_{T_i}\}$. For the revoked identities contained in $\{ID_{T_i}\}$, $\perp_i$ is returned.

Oracle $\mathcal{O}_\sigma(ID_{T_r},\ M)$. Upon being queried with a challenge M and a non-revoked identity ID_{T_r}, $\mathcal{O}_\sigma$ returns a signature σ such that $\sigma^e = \mathcal{F}\left(ID_{T_r},\ M\right)$. Otherwise, $\perp_i$ is returned.

Oracle $\mathcal{O}_p$. For a random choice of $(ID_{T_r},\ M,\ \{ID_{T_i}\})$, where the identities are not revoked, $\mathcal{O}_p$ returns the outputs of $\mathcal{O}_\sigma(ID_{T_r},\ M)$ and of $\mathcal{O}_{\Sigma_i}(ID_{T_r},\ M,\ \{ID_{T_i}\})$.

MRTDs privacy. The MRTD *privacy* should be protected at all time before the completion of augmented terminal authentication. In other words, no information regarding time or the MRTD identity should be extractable from the MRTD challenge. Hence the challenge must be completely independent from the MRTD identity, and from time too. This means that the challenge should not contain them, nor be derived from them.

Unforgeability, threshold security and robustness. The signature scheme used must be *existentially unforgeable*, to provide security against existential forgery under an adaptive chosen message attack where the adversary is allowed to query the signing oracles $\mathcal{O}_{\Sigma_i}$ and $\mathcal{O}_\sigma$. This unforgeability requirement prevents an attacker $\mathcal{A}$ from forging a valid signature σ or a valid partial signature σ_i for $\mathcal{F}\left(ID_{T_\mathcal{A}},\, M\right)$, where $ID_{T_\mathcal{A}}$ is the identity of a terminal controlled by $\mathcal{A}$.
As threshold security is assumed, the signature will have to be threshold secure and robust, as defined in Section 2.5.2.

Randomness freshness. Replaying a given challenge should happen only with negligible probability, in order to ensure the *freshness* of the authentication procedure.

Signature freshness. Terminals should not be able to reuse previously emitted signature, except with negligible probability. To enforce this, $\mathcal{F}$ is required to be second preimage resistant. This is so as to avoid that an attacker $\mathcal{A}$, upon receiving a challenge M, forges a challenge $M_\mathcal{A}$ such that

$$\mathcal{F}\left(ID_{T_\mathcal{A}},\, M\right) = \mathcal{F}\left(ID_{T_r},\, M_\mathcal{A}\right),$$

in which case $\mathcal{A}$ would be able to pass any authentication by requesting a signature on $M_\mathcal{A}$ from terminal T_r.

Revocation implications. For *revocation*, terminals that are revoked should be forbidden from participating in any authentication procedure. They should not be able to succeed in any terminal authentication request except with negligible probability.

Completeness. In the ideal case where there are no adversaries and where all participants are honest, the protocol should always succeed with overwhelming probability.

Definition 7.1 (Completeness)
A protocol for terminal authentication is said to be complete *if the following holds:*
If all participants are honest and non-revoked, the MRTD should accept the authentication procedure with overwhelming probability.

Soundness. This property exhibits the fact that T_r should not be able to cheat, except with negligible probability. To achieve that, consider the advantage of the adversary in winning some specific games. The case of passive adversaries is taken into account with the game 7.1, while game 7.2 handles the case of active adversaries. In both cases, the goal of the attacker $\mathcal{A}$ is to produce a valid signature $\sigma_\mathcal{A}$ such that the requesting MRTD will accept it. Furthermore, $\mathcal{A}$ is allowed to query oracle $\mathcal{O}_p$ for both passive and active attacks. In the case of active adversaries, $\mathcal{A}$ will additionally be allowed to query oracles $\mathcal{O}_{\Sigma_i}$ and $\mathcal{O}_\sigma$, with one restriction. The oracles can not be queried with the pair $(ID_P,\, M)$, where M is the challenge sent by the verifier MRTD and ID_P is the identity of the prover expected by the verifier MRTD. At the end of the game, $\mathcal{A}$ produces a signature $\sigma_\mathcal{A}$ and wins the game if the verifier MRTD accepts it.

Game 7.1 – Soundness game for passive adversaries (SG_{pa})

1. $\mathscr{A}$ selects t terminals to corrupt. This set is fixed.
2. The DV Key Generation and Setup are run. $\mathscr{A}$ retrieves $(Cert_{DV},\ pk)$ and for each corrupted terminal T_i, $\mathscr{A}$ also retrieves $(Cert_{T_i},\ sk_i)$.
3. $\mathscr{A}$ interacts solely with $\mathscr{O}_p$.
4. $\mathscr{A}$ selects a terminal $T_{\mathscr{A}}$ from his set of corrupted terminals. $T_{\mathscr{A}}$ will be the terminal interacting with the MRTD, hence $ID_{T_{\mathscr{A}}}$ is sent to the MRTD.
5. The MRTD generates a random challenge M and sends it back to $\mathscr{A}$.
6. $\mathscr{A}$ interacts again with $\mathscr{O}_p$.
7. $\mathscr{A}$ outputs a guess on $\sigma_{\mathscr{A}}$ and wins the game if the signature is accepted by the MRTD. In other words, $\mathscr{A}$ wins if $\sigma_{\mathscr{A}}$ is a correct signature for the pair $(ID_{T_{\mathscr{A}}},\ M)$.

Game 7.2 – Soundness game for active adversaries (SG_{aa})

1. $\mathscr{A}$ selects t terminals to corrupt. This set is fixed.
2. The DV Key Generation and Setup are run. $\mathscr{A}$ retrieves $(Cert_{DV},\ pk)$ and for each corrupted terminal T_i, $\mathscr{A}$ also retrieves $(Cert_{T_i},\ sk_i)$.
3. $\mathscr{A}$ interacts with $\mathscr{O}_p$, with $\mathscr{O}_{\Sigma_i}$ and with $\mathscr{O}_\sigma$, in a completely free manner.
4. $\mathscr{A}$ selects a terminal $T_{\mathscr{A}}$ from his set of corrupted terminals. $T_{\mathscr{A}}$ will be the terminal interacting with the MRTD, hence $ID_{T_{\mathscr{A}}}$ is sent to the MRTD.
5. The MRTD generates a random challenge M and sends it back to $\mathscr{A}$.
6. $\mathscr{A}$ interacts again with $\mathscr{O}_p$, with $\mathscr{O}_{\Sigma_i}$, and with $\mathscr{O}_\sigma$, with the restriction that the pair $(ID_{T_{\mathscr{A}}},\ M)$ can be fed neither to the oracle $\mathscr{O}_{\Sigma_i}$ nor to the oracle $\mathscr{O}_\sigma$.
7. $\mathscr{A}$ outputs a guess on $\sigma_{\mathscr{A}}$ and wins the game if the signature is accepted by the MRTD. In other words, $\mathscr{A}$ wins if $\sigma_{\mathscr{A}}$ is a correct signature for the pair $(ID_{T_{\mathscr{A}}},\ M)$.

The advantage of $\mathscr{A}$ is defined as the probability that $\mathscr{A}$ has to win the previous games: $Adv^{\mathscr{A}}_{SG_{pa}}$ for passive adversaries, and $Adv^{\mathscr{A}}_{SG_{aa}}$ for active adversaries.

Definition 7.2 (Soundness)
A protocol for terminal authentication is sound *against passive adversaries if $\mathscr{A}$ has a negligible advantage in the game 7.1:*

$$Adv^{\mathscr{A}}_{SG_{pa}} \in O(1/n).$$

It is sound *against active adversaries if $\mathscr{A}$ has a negligible advantage in the game 7.2:*

$$Adv^{\mathscr{A}}_{SG_{aa}} \in O(1/n).$$

Proactive security. Recall that proactive security is defined in Section 2.5.2. In this case, the attacker is allowed to corrupt more than t terminals, as long as no more than t terminals are corrupted, and hence revoked, before a secret resharing.

Definition 7.3 (Global security)
A protocol for terminal authentication is considered secure *if it is complete, sound, and if it achieves all previous security requirements.*

Security Proof

In order to achieve authentication, terminals have to provide undeniable evidence of their authenticity and non revocation status. This section will prove that the security requirements for the protocols depicted in 7.5 and in 7.6 are all met.

It is easy to see that the challenge M is indeed picked randomly, independently from time and from the MRTD identity. Hence the *privacy* requirement is provided.

The underlying threshold signature scheme used is the RSA threshold signature from [Sho00]. This guarantees *existential unforgeability*, *threshold security*, and *robustness*.

The probability that a given challenge is replayed against a specific terminal identity is:

$$\frac{1}{\left|\mathbb{Z}_n^*\right|} = \frac{1}{\varphi(n)} = \frac{1}{4\tilde{n}} \in O(1/n).$$

As this probability is negligible, the *randomness freshness* is ensured.

The signature scheme unforgeability requirement, combined with the second preimage resistance of the hash function $\mathscr{F}$, enables *signature freshness*. Indeed, reusing a previously emitted signature would imply one of the following three cases:

1. That the randomness freshness requirement failed, which happens with negligible probability as seen above.
2. That the adversary $\mathscr{A}$ was able to obtain an identical hash function $\mathscr{F}$ output from two different inputs, where one of the inputs is fixed and the other one is selected by $\mathscr{A}$. This case breaks the second preimage resistance of $\mathscr{F}$.
3. That $\mathscr{A}$ was able to obtain the same signature for two different $\mathscr{F}$ outputs. This case breaks the unforgeability security of the signature scheme.

As concerns *revocation*, any revoked terminal $T_{\mathscr{A}}$ will be completely ignored by all other honest terminals. These latter terminals will refuse to reply to any partial signatures requests from $T_{\mathscr{A}}$, and they will also avoid contacting $T_{\mathscr{A}}$ for any partial signature requests. Hence, revoked terminals are isolated. Their success probability in passing a terminal authentication procedure is limited to guessing the correct signature for a given challenge. Due to the signature freshness security property, this probability is negligible ($O(1/n)$).

Lemma 7.1 (Completeness)
Assuming an honest MRTD, honest non-revoked terminals, and a robust threshold signature scheme, the protocol depicted in Protocol 7.3 is complete *with overwhelming probability.*

Proof
As all participants are assumed to be honest and non-revoked, T_r will pass the revocation check from the contacted group of t non-revoked terminals T_i. The terminals T_i will be able to compute their partial signatures and their corresponding verification proofs, if $\mathscr{F}_{T_r}\left(ID_{T_r},\ M\right) \in \mathbb{Z}_n^*$. As T_i are also honest, their verification proofs and partial signatures will both be correct. Due to the robustness of the threshold signature scheme, T_r will succeed to verify their partial signatures and will correctly compute the general signature σ, using these t valid partial signatures and its own partial signature. The probability that $\mathscr{F}_{T_r}\left(ID_{T_r},\ M\right) \in \mathbb{Z}_n^*$ is equal to:

$$Pr\left[\mathscr{F}_{T_r}\left(ID_{T_r},\ M\right) \in \mathbb{Z}_n^*\right] = \frac{\varphi(n)}{n}.$$

Hence the verification check by the MRTD will always succeed with overwhelming probability, which completes the proof. ■

Corollary 7.2 (Completeness)
Assuming an honest MRTD, honest non-revoked terminals, and a robust threshold signature scheme, Protocol 7.5 and Protocol 7.6 are complete *with overwhelming probability.*

Proof
For Protocol 7.5, the proof is straightforward from Lemma 7.1 as Protocol 7.3 is a generalization of Protocol 7.5. As for Protocol 7.6, the robustness property of the threshold signature scheme ensures that the combined general signature will be valid, as all partial signatures are also valid. Hence it is also straightforward from Lemma 7.1 that Protocol 7.6 is complete. ■

Lemma 7.3 (Soundness)

Assuming threshold security and existential unforgeability for the threshold signature scheme used, accurate revocation, and a second preimage resistant full domain hash $\mathscr{F}$, the protocol depicted in Protocol 7.3 is sound *against passive and active adversaries.*

Proof

In order to achieve this proof, its transposition will be proven: if an adversary $\mathscr{A}$ is able to win in games SG_{pa} or SG_{aa}, depicted respectively in game 7.1 and game 7.2, then $\mathscr{A}$ is able to break one of the lemma assumptions.

Let us first consider the case of active adversaries playing the game SG_{aa}.
$\mathscr{A}$ first selects up to t terminals for corruption and retrieves their secrets after the DV Key Generation and Setup are run. Corrupted terminals are immediately identified and revoked, due to the accurate revocation assumption. A revoked terminal is implicitly considered as corrupted. Due to the unforgeability of the threshold signature scheme and as long as no more than t terminals are corrupted, the adversary will not be able to recover SK_{DV}. Furthermore, $\mathscr{A}$ is allowed to interact with the oracles $\mathscr{O}_p$, $\mathscr{O}_{\Sigma_i}$, and $\mathscr{O}_\sigma$. From it, $\mathscr{A}$ will collect signatures and partial signatures for chosen and random combinations of challenges $M_{\mathscr{A}}$ and identities ID_{T_r}. After $\mathscr{A}$ selects the terminal $T_{\mathscr{A}}$ interacting with the MRTD, the latter will challenge $\mathscr{A}$ with M. $\mathscr{A}$ will continue to interact with the oracles, without querying the MRTD challenge, and at the end of the game will output a valid signature σ on M for identity $ID_{T_{\mathscr{A}}}$.
To begin with, $\mathscr{A}$ could have received, from $\mathscr{O}_p$, either the signature on $(ID_{T_{\mathscr{A}}},\ M)$, or a signature on a different pair but with the same hash value $\widetilde{M} = \mathscr{F}\left(ID_{T_{\mathscr{A}}},\ M\right) = \mathscr{F}\left(ID_{T_r},\ M_{\mathscr{A}}\right)$. Both cases could happen with a negligible probability in $O(1/n)$. As they are negligible, it can be assumed for the rest of the proof that $\mathscr{A}$ did not receive them. If the existential unforgeability assumption is considered valid, and if $\mathscr{F}$ is indeed second preimage resistant, then $T_{\mathscr{A}}$ being revoked implies that $\mathscr{A}$ is able to produce a signature solely from the t corrupted terminals, breaking the threshold security assumption. If $T_{\mathscr{A}}$ was not revoked, it would be the accurate revocation assumption that would be failing. If the signature scheme used is now considered threshold secure and that the accurate revocation is successful, then either $\mathscr{A}$ breaks the second preimage resistance of $\mathscr{F}$ or the existential unforgeability assumption. Indeed, either $\mathscr{A}$ has produced a σ from a forged different pair $\left(ID_{T_r},\ M_{\mathscr{A}}\right)$, on which $\mathscr{F}\left(ID_{T_{\mathscr{A}}},\ M\right) = \mathscr{F}\left(ID_{T_r},\ M_{\mathscr{A}}\right)$, or $\mathscr{A}$ can be used as a black box in order to forge signatures for new choices of M, by simply challenging $\mathscr{A}$ with these new choices of M, breaking the existential unforgeability assumption.
In order to be able to produce σ while all the lemma assumptions hold, $\mathscr{A}$ would then need to know the DV secret key SK_{DV}. Hence $Adv^{\mathscr{A}}_{SG_{aa}} \in O(1/n)$.

The proof for passive adversaries playing the game SG_{pa} is identical, except that $\mathscr{A}$ is restricted to using $\mathscr{O}_p$ only. Hence, $\mathscr{A}$ is not allowed to make specific queries for his choices of $\left(ID_{T_r},\ M_{\mathscr{A}}\right)$. The main implication is that even if a passive adversary is able to break the second preimage resistance of $\mathscr{F}$, it will still need to wait for $\mathscr{O}_p$ to provide the corresponding signature. Hence $Adv^{\mathscr{A}}_{SG_{pa}} \in O(1/n)$. ■

Corollary 7.4 (Soundness)

Assuming threshold security and existential unforgeability for the threshold signature scheme used, accurate revocation and a second preimage resistant full domain hash $\mathscr{F}$, Protocols 7.5 and 7.6 are sound *against passive and active adversaries.*

Proof

For Protocol 7.5, the proof is straightforward from Lemma 7.3 as Protocol 7.3 is a generalization of Protocol 7.5. As for Protocol 7.6, if at least one of the partial signatures is invalid, the Lagrange interpolation will fail to provide a valid signature σ. Hence it is also straightforward from Lemma 7.3 that Protocol 7.6 is sound. ■

Theorem 7.5

Assuming threshold security and existential unforgeability for the robust threshold signature scheme used, accurate revocation and a second preimage resistant full domain hash $\mathscr{F}$, the protocol depicted in Protocol 7.3 is secure.

Proof

The proof is straightforward from Lemma 7.1 and from Lemma 7.3. ■

Corollary 7.6

Assuming threshold security and existential unforgeability for the robust threshold signature scheme used, accurate revocation and a second preimage resistant full domain hash $\mathscr{F}$, Protocols 7.5 and 7.6 are secure.

Proof

The proof is straightforward from Corollary 7.2 and from Corollary 7.4. ■

General Analysis

A stolen or malicious terminal will not be able to authenticate itself, nor to impersonate another valid terminal. A corrupted collaborating terminal will learn no information except that a MRTD with some random challenge has requested an authentication process. However, a corrupted terminal interacting with a MRTD will be granted access to the MRTD sensitive data if the corrupted terminal behaves honestly. As long as at most t terminals are corrupted, the DV secret key used to authenticate terminals remains protected.

In the case that the requesting terminal is non-revoked and compromised, the adversary could gain access to sensitive data from the MRTDs that it encounters. However in order to do so, the requesting terminal will have to collaborate honestly with the other terminals. This adversarial behavior can be mitigated by monitoring the network and making sure that terminals only communicate with other known terminals.

When the requesting terminal is honest and some collaborating terminals are corrupted and non-revoked, these terminals will be easily identified if they fail to provide valid verification proofs on their partial signatures. Moreover, the adversary will only learn contents of challenges without being able to link them to the MRTDs that generated them.

Proactive security can be achieved by frequently renewing the global secret of the threshold signature scheme. This can be done efficiently by resharing the same secret by means of sharing the "secret" value '0' and adding the obtained partial secrets to the previous ones. This technique can be easily explained by the Lagrange interpolation. Assume the general secret is contained in $f(0)$ and that another function g, with $g(0) = 0$ is shared and added to the previous secret shares. The resulting addition will form another function $\hat{f}$ such that $\hat{f}(0) = f(0)$. This method reduces the threat of terminal keys being exposed. In order to compromise the general secret key, an adversary would have to obtain $t+1$ key shares in the same time frame. This allows DV certificates to protect their general secret used for threshold signature throughout their entire period of validity. Notice that this step is highly efficient if performed by the DV. The DV would generate the additional secret key shares and distribute them to their corresponding terminal. Verification keys will also have to be redistributed to every participant. However, this can also be achieved without the DV by using secure multiparty computation techniques [DK01], although this will imply a loss of efficiency.

In conclusion, a stolen terminal will not be able to authenticate itself. A corrupted collaborating terminal will learn no information except that a MRTD has requested an authentication process. However, a corrupted requesting terminal interacting with a MRTD will be granted access to the MRTD sensitive data if the terminal behaves honestly with the other collaborating terminals. As long as at most t terminals are corrupted, the secret key used to authenticate terminals remains protected. Furthermore, in case of proactive security, the leakage of the secret key can be achieved only if at least $t+1$ key shares are compromised within the same time frame of a resharing phase. These security properties are desirable as they improve the current state of the EAC. By lowering the trust in terminals, the level of trust in the DV is increased. This is an acceptable change as DVs are less exposed than terminals.

Remarks

Where the requesting terminal T_r is both corrupted and not revoked remains an open problem. SK_{DV} would remain protected but the adversary would be able to collect data from passports. A potential mitigation would be to perform a continuous network analysis to check if terminals are connected with illegitimate entities.

It is important to note that the solution presented in this chapter does not solve the problem of a malicious DV. Indeed the DV can lower the requirements for its terminal in order to retrieve as much data as possible among the people who transit its borders. Hence if a country is subject to privacy infringement or if a country is known to disrespect the privacy of people, DVs from that country should be forbidden to access sensitive information contained in MRTDs. The only alternative solutions to disrespectful DVs and/or countries are expensive and cumbersome. For instance, a possibility would be the solution provided by Buchmann and Baier in [BB13], where the authentication and revocation of terminals are enabled by a home server from the MRTD country.

Chapter 8

Conclusion

Due to a high level of public interest in privacy and e-services, this thesis focused on:

- improving cryptographic primitives necessary for enhancing privacy protection; and
- surveying and improving the standards for Machine Readable Travel Documents.

In relation to the former, two primitives were studied and improved: set membership and range proofs. Moreover, solutions for range proofs were provided in the interactive and non-interactive communication models. For the latter, the ICAO and the EAC standards were surveyed, and improvements were elaborated.

For set membership proofs, a first solution was constructed based on the Boneh-Boyen signature scheme. The scheme relies on proving that a signature for the committed secret element is known to the prover. This thesis argued that other signature schemes could be employed and provided an example with the Camenisch-Lysyanskaya signature scheme, although slightly less efficient. This thesis then provided a general explanation of how to build a set membership proof based on any secure signature scheme. Furthermore, this thesis showed that cryptographic accumulators could also replace signature schemes in the construction of set membership proofs. The most efficient secure protocol for set membership proof is currently the one based on the Boneh-Boyen signature scheme. A variant of this scheme was proposed by Arfaoui et al. in [ALT+15a], which attempted with limited success, to reduce the computation complexity of verifying the Boneh-Boyen signatures.

Regarding interactive range proofs, this thesis improved range decomposition methods and combined them with a proof of signature knowledge. This led to efficient protocols, where elements are first decomposed and their digits are then proven with the set membership proofs developed in this thesis. The interactive range proofs schemes presented in this thesis reached the efficient asymptotical communication complexity of $O\left(\frac{\kappa}{\log\kappa - \log\log\kappa}\right)$, where κ is the security parameter. Furthermore, the author of this thesis conjectures that this communi-

cation complexity is an asymptotical lower bound, with current commitment schemes. This bound might be lowered by using different security requirements, especially for commitments. A possible solution might be achieved with commitments built on unconventional security requirements. For instance, if a secret element is outside a given range, its commitment would no longer be hiding and the secret would be leaked.

It is important to keep in mind that the choice of protocol for interactive range proofs is dependent on the range size and the security desired. If the range size is larger than 2^{256} for 1024 bits group elements or 2^{1642} for 20 bytes group elements, then the Groth positivity test method [Gro05] is more efficient than the schemes presented in this thesis, regarding the communication complexity. If it is acceptable to lower the security requirements from zero-knowledge to witness indistinguishability, then for a range size that is larger than 2^{32}, Groth binary decomposition [Gro11] is supposed to be more efficient, although it induces a 7 rounds protocol. For range sizes smaller than 2^{32} or if zero-knowledge is preferred, then the sumset based range proof presented in this thesis remains the most efficient and secure choice. Last but not least, for very small ranges (for instance less than 32 elements), the set membership proof primitives presented in this thesis are more efficient.

For efficient non-interactive range proofs without random oracles, this thesis proved the insecurity of the first attempted protocol, which was elaborated by Yuen et al. in [YHM+09]. This thesis then presented a construction of a flexible solution based on the sumset decomposition, on the Λ-PKE knowledge assumption, on a lifted version of the BBS cryptosystem, on a Hadamard product argument, and on the Lipmaa permutation argument. The protocol resulting from this solution achieves a minimal communication complexity of 35 group elements, in the binary sumset decomposition case. Although this protocol was the most efficient solution when published, the current state of the art is provided by Lipmaa in [Lip14b, Lip16], which achieves a constant communication complexity of 11 group elements.

Concerning MRTDs, this thesis explained the threats linked to the hardware choice of RFID chips. This thesis surveyed the ICAO standard and explained the weaknesses of its terminal authentication procedure, which is based on the Basic Access Control (BAC) protocol. This thesis surveyed both versions of the EAC standard (EACv1 and EACv2), explained the improvements achieved by these standards, and their remaining drawbacks, notably the lack of privacy control on the data contained in the MRTDs, the weaknesses of the terminal authentication protocol, and more importantly, the terminal revocation problem.

The outcome of the survey on the ICAO standard presented in this thesis, demonstrated that the BAC should be replaced by its equivalent from the EACv2 standard, namely the Password Authenticated Connection Establishment (PACE). This recommendation was adopted in February 2013 by the ICAO working group ISO/IEC JTC1 SC17 WG3 [ICAO13]. Furthermore, this thesis recommended the introduction of an RFID switch in MRTDs, in order to easily enable privacy protection. However, it has been mentioned to the author of this thesis by the German *Federal Office for Information Security* (BSI), that it would be too difficult to obtain the

approval for hardware modifications regarding MRTDs. Hence, terminal revocation remains based on the expiration date of their certificate, while MRTDs only have a poor approximation of the current date. In an attempt to reduce this problem, this thesis recommended to increase the updates of the date approximation of MRTDs. Furthermore, this thesis elaborated a better solution to resolve the terminal revocation problem. This solution requires terminals to collaborate in order to authenticate themselves, which solves the threat of an isolated rogue terminal.

Open Problems. Remark that the set membership and range proofs presented in this thesis are all dependent on the discrete logarithm computational hardness assumption. Hence they are not quantum secure. Constructing set membership and range proofs that are quantum secure remains an open problem. A potential solution might be obtained from latices. A further remaining open problem consists of proving the asymptotical lower bound $O\left(\frac{\kappa}{\log\kappa - \log\log\kappa}\right)$ for the computational complexity of range proofs. Regarding MRTDs, the standardization authorities are aware of the solutions presented in this thesis and their implementation is hindered by a lack of political will.

Appendix A

Proof of Knowledge of a Camenisch-Lysyanskaya Signature

The proof of knowledge of a Camenisch-Lysyanskaya signature suggested in Figure 3 of [CL02b], is detailed here. This proof of knowledge is needed in Protocol 3.2 in Section 3.4. The objective of the proof of knowledge is the following:

$$PK\{(x, r, s, e, v) : C_x = g^x h^r \ \wedge \ \mathsf{Verify}_{(\tilde{n},a,b,\tilde{c})}(x, s, e, v) = 1\},$$

where the predicate Verify is the verification algorithm of the Camenisch-Lysyanskaya signature, as defined in Section 3.4.

Common Input:	g, h, commitment C_x, $\widetilde{g}$ and $\widetilde{h}$ for a commitment scheme modulo $\widetilde{n}$, public key $(\widetilde{n}, a, b, \widetilde{c})$ of the Camenisch-Lysyanskaya signature scheme, commitments C_v and C_w, and security parameters ℓ_e, ℓ_m, ℓ_n.
Prover Input[a]**:**	x, r such that $C_x = g^x h^r \bmod n$, s, e, v, w, r_w such that $v^e = a^x b^s c \bmod \widetilde{n}$, such that $C_v = v\widetilde{g}^w$ and $C_w = \widetilde{g}^w \widetilde{h}^{r_w}$.
$P \xrightarrow{t_1, t_2, t_3} V$	• Prover picks uniformly at random values of length ℓ_n: $r_x,\ r_\rho,\ r_s,\ r_e,\ r_\delta,\ r_\gamma$, • Prover sends $t_1 \leftarrow g^{r_x} h^{r_\rho}$, $t_2 \leftarrow C_w^{r_e} \widetilde{g}^{-r_\delta} \widetilde{h}^{-r_\gamma}$, and $t_3 \leftarrow C_v^{r_e} \widetilde{g}^{-r_\delta} a^{-r_x} b^{-r_s}$.
$P \xleftarrow{c} V$	• Verifier picks uniformly at random c of length ℓ_n,
$P \xrightarrow{s_x, s_r, s_\xi, s_\varepsilon, s_\delta, s_\gamma} V$	• Prover sends $s_x \leftarrow r_x - cx,\ s_r \leftarrow r_\rho - cr$, $s_\xi \leftarrow r_s - cs,\ s_\varepsilon \leftarrow r_e - ce$, $s_\delta \leftarrow c\delta - r_\delta,$ where $\delta = we$ and $s_\gamma \leftarrow c\gamma - r_\gamma,$ where $\gamma = r_w e$. • Verifier checks that $t_1 \stackrel{?}{=} C_x^c g^{s_x} h^{s_r}$, that $t_2 \stackrel{?}{=} C_w^{s_\xi} \widetilde{g}^{s_\delta} \widetilde{h}^{s_\gamma}$, and that $t_3 \stackrel{?}{=} C_v^{s_\varepsilon} \cdot a^{-s_x} b^{-s_\xi} \widetilde{c}^c \cdot \widetilde{g}^{s_\delta}$.

Protocol A.1 – Proof of knowledge of a Camenisch-Lysyanskaya Signature

[a] The prover also needs to additionally run, in parallel, two range proofs for $x \in \left(2^{\ell_m - 1}, 2^{\ell_m}\right)$ and for $e \in \left(2^{\ell_e - 1}, 2^{\ell_e}\right)$.

Appendix B

Proof of Knowledge of a Committed Accumulated Element

The proof of knowledge, used in Section 3.5, is specified here. Its goal is to prove that a committed value is contained in a given accumulator. This protocol is based on the results of Camenisch and Lysyanskaya from [CL02a]. The difference between the following proof of knowledge and their result is that the committed element is not necessarily a prime number corresponding to their requirements. Hence a mapping is needed. The objective of the proof of knowledge is the following:

$$
\begin{aligned}
PK\{(\sigma, r, e_\sigma, a_\sigma, r_e, r_1, r_2) : \; & C = g^\sigma h^r \;\wedge\; C_e = \widetilde{g}^{e_\sigma} \widetilde{h}^{r_e} \pmod{\widetilde{n}} \;\wedge \\
& C_e = (\widetilde{g}^{2^k})^\sigma \widetilde{g}^{a_\sigma} \widetilde{h}^{r_e} \pmod{\widetilde{n}} \;\wedge\; R^{e_\sigma} = \widetilde{g}^{r_1 e_\sigma} \widetilde{h}^{r_2 e_\sigma} \pmod{\widetilde{n}} \;\wedge \\
& v = W^{e_\sigma} \widetilde{h}^{-r_1 e_\sigma} \pmod{\widetilde{n}} \;\wedge\; a_\sigma \in [-2^{k-1}, 2^{k-1}]\}
\end{aligned}
$$

Common Input:	g, h, commitment C, set Φ,
	$\widetilde{n}$, accumulator v, $\widetilde{g}$, $\widetilde{h}$, set §,
	auxiliary commitments W, R, C_e.
Prover Input:	σ, r such that $C = g^{\sigma} h^{r}$ and $\sigma \in \Phi$,
	$r_1, r_2, r_e, e_\sigma, w_\sigma, a_\sigma$ such that $(e_\sigma, w_\sigma) \in$ §, $e_\sigma = \sigma 2^{k} + a_\sigma$,
	such that $W = w_\sigma \widetilde{h}^{r_1} \pmod{\widetilde{n}}$, $R = \widetilde{g}^{r_1} \widetilde{h}^{r_2} \pmod{\widetilde{n}}$,
	and $C_e = \widetilde{g}^{e_\sigma} \widetilde{h}^{r_e} \pmod{\widetilde{n}}$.
Verifier Input:	$\widetilde{p}$, $\widetilde{q}$ such that $\widetilde{n} = (2\widetilde{p}+1)(2\widetilde{q}+1)$.
$P \xrightarrow{t_1,t_2,t_3,t_4,t_5} V$	• Prover picks $r_\sigma, r_\rho \in_R \mathbb{Z}_q$[a],
	picks $r_\varepsilon \in_R \left(-B2^{k'+k''},\ B2^{k'+k''}\right)$,
	$r_\xi \in_R \left(-\lfloor \widetilde{n}/4 \rfloor 2^{k'+k''},\ \lfloor \widetilde{n}/4 \rfloor 2^{k'+k''}\right)$,
	$r_\alpha \in_R \left(-2^{\tilde{k}+k'+k''},\ 2^{\tilde{k}+k'+k''}\right)$,
	$r_\delta, r_\gamma \in_R \left(-\lfloor \widetilde{n}/4 \rfloor B2^{k'+k''},\ \lfloor \widetilde{n}/4 \rfloor B2^{k'+k''}\right)$,
	sends $t_1 \leftarrow g^{r_\sigma} h^{r_\rho}$, $t_2 \leftarrow \widetilde{g}^{r_\varepsilon} \widetilde{h}^{r_\xi}$, $t_3 \leftarrow \left(\widetilde{g}^{2^k}\right)^{r_\sigma} \widetilde{g}^{r_\alpha} \widetilde{h}^{r_\xi}$,
	$t_4 \leftarrow R^{r_\varepsilon} \widetilde{g}^{-r_\delta} \widetilde{h}^{-r_\gamma}$ and $t_5 \leftarrow W^{r_\varepsilon} \widetilde{h}^{-r_\delta}$.
$P \xleftarrow{c} V$	• Verifier picks and sends $c \in_R \{0,1\}^{k'}$,
$P \xrightarrow{s_\sigma,s_\rho,s_\varepsilon,s_\xi,s_\alpha,s_\delta,s_\gamma} V$	• Prover sends $s_\sigma \leftarrow r_\sigma - c\sigma \pmod{q}$, $s_\rho \leftarrow r_\rho - cr \pmod{q}$,
	sends $s_\varepsilon \leftarrow r_\varepsilon - ce_\sigma$, $s_x i \leftarrow r_\xi - cr_e$, $s_\alpha \leftarrow r_\alpha - ca_\sigma$,
	$s_\delta \leftarrow c\delta - r_\delta$, where $\delta = r_1 e_\sigma$ and
	$s_\gamma \leftarrow c\gamma - r_\gamma$, where $\gamma = r_2 e_\sigma$.
	• Verifier checks that $t_1 \stackrel{?}{=} C^c g^{s_\sigma} h^{s_\rho}$, that $t_2 \stackrel{?}{=} C_e^c \widetilde{g}^{s_\varepsilon} \widetilde{h}^{s_\xi}$,
	that $t_3 \stackrel{?}{=} C_e^c \left(\widetilde{g}^{2^k}\right)^{s_\sigma} \widetilde{g}^{s_\alpha} \widetilde{h}^{s_\xi}$,
	that $t_4 \stackrel{?}{=} R^{s_\varepsilon} \widetilde{g}^{s_\delta} \widetilde{h}^{s_\gamma}$, $t_5 \stackrel{?}{=} v^c W^{s_\varepsilon} \widetilde{h}^{s_\delta}$,
	and that $s_\alpha \in [-2^{k-2}, 2^{k-2}]$.

Protocol B.1 – Proof of knowledge of a committed accumulated element $\sigma \in \Phi$

[a] Here, q is the order of the Pedersen commitment group.

Appendix C

Computational Complexity Comparisons of Interactive Range Proofs

Schemes	*Computational Complexity*	
	Prover	*Verifier*
CCs_AND (Protocol 4.2)	$(6\ell+2)$ exp., 2ℓ pairings	$(4\ell+7)$ exp., 4ℓ pairings
Lipmaa [Lip03] (Sum of 4 squares)	36 exp. + $O(k^2)$ op.	36 exp.
Boudot [Bou00] (Square + CFT [CFT98b])	29 exp.	24 exp.
Groth [Gro05] (Sum of 3 squares)	28 exp. + $O(k^2)$ op.	28 exp.
Scemama [Sce09] (Square + CFT [CFT98b])	27 exp.	21 exp.
CCs_AND_Arfaoui (AND composition with Protocol 4.4)	$(6\ell+2)$ exp.	$(4\ell+6)$ exp.
Groth [Gro11] (binary decomposition of commitments of commitments)	$O(k^{2/3})$ exp., $O(k^{2/3})$ pairings	$O(k^{1/3})$ exp., $O(k^{1/3})$ pairings
Sumset based range proof (Protocol 4.5)	$(3\ell+5)$ exp.	$(2\ell+5)$ exp.

Figure C.1 – Complexity comparison for range $[A,B]$, with $k=\log_2 B$, $u^{\ell-1} < B+1-A < u^{\ell}$, and 128 bit security. Complexities are provided in terms of exponentiations (exp.) and pairings.

Bibliography

[AA08] Mohamed Abid and Hossam Afifi. Secure e-passport protocol using elliptic curve diffie-hellman key agreement protocol. In *Proccedings of the Fourth International Conference on Information Assurance and Security, IAS 2008, Napoli, Italy*, pages 99–102, 2008.

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity. A modern approach*. Cambridge: Cambridge University Press, 2009.

[ACP09] Michel Abdalla, Céline Chevalier, and David Pointcheval. Smooth projective hashing for conditionally extractable commitments. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA*, pages 671–689, 2009.

[AFG+10] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA*, pages 209–236, 2010.

[ALT+15a] Ghada Arfaoui, Jean-François Lalande, Jacques Traoré, Nicolas Desmoulins, Pascal Berthomé, and Saïd Gharout. A practical set-membership proof for privacy-preserving NFC mobile ticketing. *CoRR*, abs/1505.03048, 2015.

[ALT+15b] Ghada Arfaoui, Jean-François Lalande, Jacques Traoré, Nicolas Desmoulins, Pascal Berthomé, and Saïd Gharout. A practical set-membership proof for privacy-preserving NFC mobile ticketing. *PoPETs*, 2015(2):25–45, 2015.

[AMOR14] Gora Adj, Alfred Menezes, Thomaz Oliveira, and Francisco Rodríguez-Henríquez. Computing discrete logarithms in $\mathbb{F}_{3^{6\ldots137}}$ and $\mathbb{F}_{3^{6\ldots163}}$ using magma. In *Arithmetic of Finite Fields - 5th International Workshop, WAIFI 2014, Gebze, Turkey*, pages 3–22, 2014.

[AO05] Gildas Avoine and Philippe Oechslin. RFID traceability: A multilayer problem. In *Financial Cryptography and Data Security, 9th International Conference, FC 2005, Roseau, The Commonwealth of Dominica*, pages 125–140, 2005.

[AWSM07] Man Ho Au, Qianhong Wu, Willy Susilo, and Yi Mu. Compact e-cash from bounded accumulator. In *Topics in Cryptology - CT-RSA 2007, The Cryptographers' Track at the RSA Conference 2007, San Francisco, CA, USA*, pages 178–195, 2007.

[Bay13] Stephanie Bayer. *Practical zero-knowledge Protocols based on the discrete logarithm Assumption*. PhD thesis, Department of Computer Science, University College London, London, 2013.

[BB04] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland*, pages 56–73, 2004.

[BB13] Nicolas Buchmann and Harald Baier. Towards a more secure and scalable verifying PKI of eMRTD. In *Public Key Infrastructures, Services and Applications - 10th European Workshop, EuroPKI 2013, Egham, UK*, pages 102–118, 2013.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology Conference, Santa Barbara, California*, pages 41–55, 2004.

[BCDvdG87] Ernest F. Brickell, David Chaum, Ivan Damgård, and Jeroen van de Graaf. Gradual and verifiable release of a secret. In *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA*, pages 156–166, 1987.

[BCKL08] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. P-signatures and noninteractive anonymous credentials. In *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA*, pages 356–374, 2008.

[BCM05] Endre Bangerter, Jan Camenisch, and Ueli M. Maurer. Efficient proofs of knowledge of discrete logarithms and representations in groups with hidden order. In *Public Key Cryptography - PKC 2005, 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland*, pages 154–171, 2005.

[BCPR14] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. On the existence of extractable one-way functions. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA*, pages 505–514, 2014.

[BCV15] Olivier Blazy, Céline Chevalier, and Damien Vergnaud. Non-interactive zero-knowledge proofs of non-membership. In Kaisa Nyberg, editor, *Topics in Cryptology, CT-RSA 2015*, volume 9048 of *Lecture Notes in Computer Science*, pages 145–164. Springer International Publishing, 2015.

[BDFK12] Jens Bender, Özgür Dagdelen, Marc Fischlin, and Dennis Kügler. The PACE|AA protocol for machine readable travel documents, and its security. In *Financial Cryptography and Data Security - 16th International Conference, FC 2012, Kralendijk, Bonaire*, pages 344–358, 2012.

[BDG+13] Nir Bitansky, Dana Dachman-Soled, Sanjam Garg, Abhishek Jain, Yael Tauman Kalai, Adriana López-Alt, and Daniel Wichs. Why "Fiat-Shamir for Proofs" lacks a proof. In *TCC*, pages 182–201, 2013.

[BdM93] Josh Cohen Benaloh and Michael de Mare. One-way accumulators: A decentralized alternative to digital sinatures (extended abstract). In *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of Cryptographic Techniques, Lofthus, Norway*, pages 274–285, 1993.

[BDPA13] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece*, pages 313–314, 2013.

[BFK09] Jens Bender, Marc Fischlin, and Dennis Kügler. Security analysis of the pace key-agreement protocol. In *Information Security, 12th International Conference, ISC 2009, Pisa, Italy*, pages 33–48, 2009.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, Chicago, Illinois, USA*, pages 103–112, 1988.

[BG92] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA*, pages 390–420, 1992.

[BG97] Mihir Bellare and Shafi Goldwasser. Verifiable partial key escrow. In *CCS '97, Proceedings of the 4th ACM Conference on Computer and Communications Security, Zurich, Switzerland*, pages 78–91, 1997.

[BG13] Stephanie Bayer and Jens Groth. Zero-knowledge argument for polynomial evaluation with application to blacklists. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece*, pages 646–663, 2013.

[BGI+14] Cyril Bouvier, Pierrick Gaudry, Laurent Imbert, Hamza Jeljeli, and Emmanuel Thomé. Discrete logarithms in GF(p) — 180 digits. https://listserv.nodak.edu/cgi-bin/wa.exe?A2=NMBRTHRY;615d922a.1406, 2014.

[BK08] Abklärungen über die datenauslesung auf distanz beim biometrischen pass. Bundesamt für Kommunikation BAKOM, November 2008.

http://www.schweizerpass.admin.ch/etc/medialib/data/passkampagne/e-paesse.Par.0004.File.tmp/Messbericht_Bakom.pdf.

[Bla79] G. R. Blakley. Safeguarding cryptographic keys. In *Proceedings of the National Computer Conference, 1979, American Federation of Information Processing Societies - Conference Proceedings, Vol. 48, AFIPS Press, Montvale, New Jersey*, pages 313–317, 1979.

[Bla97] Kelly Black. Classroom Note: Putting Constraints in Optimization for First-Year Calculus Students. *SIAM Rev.*, 39(2):310–312, 1997.

[BLR12] Rima Belguechi, Patrick Lacharme, and Christophe Rosenberger. Enhancing the privacy of electronic passports. *IJITM*, 11(1/2):122–137, 2012.

[Blu81] Manuel Blum. Coin flipping by telephone. In *Advances in Cryptology: A Report on CRYPTO 81, CRYPTO 81, IEEE Workshop on Communications Security, Santa Barbara, California, USA*, pages 11–15, 1981.

[BMRV00] Harry Buhrman, Peter Bro Miltersen, Jaikumar Radhakrishnan, and Srinivasan Venkatesh. Are bitvectors optimal? In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, Portland, OR, USA*, pages 449–458, 2000.

[BN05] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography, 12th International Workshop, SAC 2005, Kingston, ON, Canada*, pages 319–331, 2005.

[BNF12] Nasima Begum, Toru Nakanishi, and Nobuo Funabiki. Efficient proofs for CNF formulas on attributes in pairing-based anonymous credential system. In *Information Security and Cryptology - ICISC 2012 - 15th International Conference, Seoul, Korea*, pages 495–509, 2012.

[Bol03] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In *Public Key Cryptography - PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA*, pages 31–46, 2003.

[Bou00] Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium*, pages 431–444, 2000.

[BP97] Niko Barić and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany*, pages 480–494, 1997.

[BP04] Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology Conference, Santa Barbara, California*, pages 273–289, 2004.

[BPW12] David Bernhard, Olivier Pereira, and Bogdan Warinschi. How not to prove yourself: Pitfalls of the fiat-shamir heuristic and applications to helios. In *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China*, pages 626–643, 2012.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA*, pages 62–73, 1993.

[BR96] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures - how to sign with RSA and Rabin. In *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain*, pages 399–416, 1996.

[BS03] Adam Barnett and Nigel P. Smart. Mental poker revisited. In *Cryptography and Coding, 9th IMA International Conference, Cirencester, UK, December 16-18, 2003, Proceedings*, pages 370–383, 2003.

[BSI06] BSI Bundesamt für Sicherheit in der Informationstechnik. Advanced security mechanisms for machine readable travel documents – extended access control (EAC). Technical report, Federal Office for Information Security, 53133 Bonn, Germany, 2006. Technical Guideline TR-03110, Version 1.00.

[BSI08a] BSI Bundesamt für Sicherheit in der Informationstechnik. Advanced security mechanisms for machine readable travel documents – Extended Access Control (EAC). Technical report, Federal Office for Information Security, 53133 Bonn, Germany, 2008. Technical Guideline TR-03110, Version 1.11.

[BSI08b] BSI Bundesamt für Sicherheit in der Informationstechnik. Advanced security mechanisms for machine readable travel documents – Extended Access Control (EAC), password authenticated connection establishment (PACE), and restricted identification (RI). Technical report, Federal Office for Information Security, 53133 Bonn, Germany, 2008. Technical Guideline TR-03110, Version 2.0.

[BSI09a] BSI Bundesamt für Sicherheit in der Informationstechnik. Advanced security mechanisms for machine readable travel documents – Extended Access Control (EAC), password authenticated connection establishment (PACE), and restricted identification (RI). Technical report, Federal Office for Information Security, 53133 Bonn, Germany, 2009. Technical Guideline TR-03110, Version 2.01.

[BSI09b] BSI Bundesamt für Sicherheit in der Informationstechnik. Pkis for machine readable travel documents – protocols for the management of certificates and CRLs. Technical report, Federal Office for Information Security, 53133 Bonn, Germany, 2009. Technical Guideline TR-03129, Version 1.10.

[BSI15a] BSI Bundesamt für Sicherheit in der Informationstechnik. Advanced security mechanisms for machine readable travel documents and eIDAS token – part 1, eMRTDs with BAC/PACEv2 and EACv1. Technical report, Federal Office for Information Security, 53133 Bonn, Germany, 2015. Technical Guideline TR-03110-1, Version 2.20.

[BSI15b] BSI Bundesamt für Sicherheit in der Informationstechnik. Advanced security mechanisms for machine readable travel documents and eIDAS token – part 2, protocols for electronic identification, authentication and trust services (eIDAS) connection establishment (PACE), and restricted identification (RI). Technical report, Federal Office for Information Security, 53133 Bonn, Germany, 2015. Technical Guideline TR-03110-2, Version 2.20.

[BSI15c] BSI Bundesamt für Sicherheit in der Informationstechnik. Advanced security mechanisms for machine readable travel documents and eIDAS token – part 3, common specifications. Technical report, Federal Office for Information Security, 53133 Bonn, Germany, 2015. Technical Guideline TR-03110-3, Version 2.20.

[BSI15d] BSI Bundesamt für Sicherheit in der Informationstechnik. Advanced security mechanisms for machine readable travel documents and eIDAS token – part 4, applications and document profiles. Technical report, Federal Office for Information Security, 53133 Bonn, Germany, 2015. Technical Guideline TR-03110-3, Version 2.20.

[BSMP91] Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Noninteractive zero-knowledge. *SIAM J. Comput.*, 20(6):1084–1118, 1991.

[BT09] Claude Barral and Assia Tria. Fake fingers in fingerprint recognition: Glycerin supersedes gelatin. In *Formal to Practical Security - Papers Issued from the 2005-2008 French-Japanese Collaborations*, pages 57–69, 2009.

[CCJT13] Sébastien Canard, Iwen Coisel, Amandine Jambert, and Jacques Traoré. New results for the practical use of range proofs. In *Public Key Infrastructures, Services and Applications - 10th European Workshop, EuroPKI 2013, Egham, UK*, pages 47–64, 2013.

[CCs08] Jan Camenisch, Rafik Chaabouni, and abhi shelat. Efficient protocols for set membership and range proofs. In *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia*, pages 234–252, 2008.

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Extractable perfectly one-way functions. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, pages 449–460, 2008.

[CDM00] Ronald Cramer, Ivan Damgård, and Philip D. MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In *Public Key Cryptography, Third International Workshop on Practice and Theory in Public Key Cryptography, PKC 2000, Melbourne, Victoria, Australia*, pages 354–373, 2000.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA*, pages 174–187, 1994.

[CDvdG87] David Chaum, Ivan Damgård, and Jeroen van de Graaf. Multiparty computations ensuring privacy of each party's input and correctness of the result. In *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA*, pages 87–119, 1987.

[CFT98a] Agnes Hui Chan, Yair Frankel, and Yiannis Tsiounis. Easy come - easy go divisible cash. In *EUROCRYPT*, pages 561–575, 1998.

[CFT98b] Agnes Hui Chan, Yair Frankel, and Yiannis Tsiounis. Easy come - easy go divisible cash. http://www.ccs.neu.edu/home/yiannis/pubs.html, May 1998. Updated version with corrections on the Range Bounded Commitment protocol.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA*, pages 209–218, 1998.

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

[CGS97] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany*, pages 103–118, 1997.

[Cha13] Rafik Chaabouni. Solving terminal revocation in EAC by augmenting terminal authentication. In *BIOSIG 2013 - Proceedings of the 12th International Conference of Biometrics Special Interest Group, Darmstadt, Germany*, pages 273–280, 2013.

[Che06] Jung Hee Cheon. Security analysis of the strong diffie-hellman problem. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia*, pages 1–11, 2006.

[CHS04] Giovanni Di Crescenzo, Javier Herranz, and Germán Sáez. Reducing server trust in private proxy auctions. In *Trust and Privacy in Digital Business, First International Conference, TrustBus 2004, Zaragoza, Spain*, pages 80–89, 2004.

[CL02a] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA*, pages 61–76, 2002.

[CL02b] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In *Security in Communication Networks, Third International Conference, SCN 2002, Amalfi, Italy*, pages 268–289, 2002.

[CLRPS06] Dario Carluccio, Kerstin Lemke-Rust, Christof Paar, and Ahmad-Reza Sadeghi. E-passport: The global traceability or how to feel like a UPS package. In *Information Security Applications, 7th International Workshop, WISA 2006, Jeju Island, Korea*, pages 391–404, 2006.

[CLs10] Rafik Chaabouni, Helger Lipmaa, and abhi shelat. Additive combinatorics and discrete logarithm based range protocols. In *Information Security and Privacy - 15th Australasian Conference, ACISP 2010, Sydney, Australia*, pages 336–351, 2010.

[CLZ12] Rafik Chaabouni, Helger Lipmaa, and Bingsheng Zhang. A non-interactive range proof with constant communication. In *Financial Cryptography and Data Security - 16th International Conference, FC 2012, Kralendijk, Bonaire*, pages 179–199, 2012.

[CNs07] Jan Camenisch, Gregory Neven, and abhi shelat. Simulatable adaptive oblivious transfer. In *Advances in Cryptology - EUROCRYPT 2007, 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain*, pages 573–590, 2007.

[Coh10] Henri Cohen. *A Course in Computational Algebraic Number Theory*. Graduate Texts in Mathematics. Springer-Verlag, 2010.

[Cor00] Jean-Sébastien Coron. On the exact security of full domain hash. In *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA*, pages 229–235, 2000.

[CP92] David Chaum and Torben P. Pedersen. Wallet databases with observers. In *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA*, pages 89–105, 1992.

[CPSV16] Michele Ciampi, Giuseppe Persiano, Luisa Siniscalchi, and Ivan Visconti. A transform for NIZK almost as efficient and general as the fiat-shamir transform without programmable random oracles. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, Proceedings, Part II*, pages 83–111, 2016.

[CR00] Certicom Research. Standards for efficient cryptography, SEC 1: Elliptic curve cryptography, September 2000. Version 1.0. http://www.secg.org/secg_docs.htm.

[Cra97] Ronald John Fitzgerald Cramer. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD thesis, University of Amsterdam, Amsterdam, 1997.

[Cre02] Giovanni Di Crescenzo. Equivocable and extractable commitment schemes. In *Security in Communication Networks, Third International Conference, SCN 2002, Amalfi, Italy*, pages 74–87, 2002.

[CV09] Rafik Chaabouni and Serge Vaudenay. The extended access control for machine readable travel documents. In *BIOSIG 2009 - Proceedings of the Special Interest Group on Biometrics and Electronic Signatures, Darmstadt, Germany*, pages 93–103, 2009.

[Dam91] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA*, pages 445–456, 1991.

[Den02] Alexander W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand*, pages 100–109, 2002.

[DF94] Yvo Desmedt and Yair Frankel. Perfect homomorphic zero-knowledge threshold schemes over any finite abelian group. *SIAM J. Discrete Math.*, 7(4):667–679, 1994.

[DF02] Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand*, pages 125–142, 2002.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DHBC09] Boris Danev, Thomas S. Heydt-Benjamin, and Srdjan Capkun. Physical-layer identification of RFID devices. In *18th USENIX Security Symposium, Montreal, Canada*, pages 199–214, 2009.

[DJ01] Ivan Damgård and Mads Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *Public Key Cryptography, 4th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2001, Cheju Island, Korea*, pages 119–136, 2001.

[DK01] Ivan Damgård and Maciej Koprowski. Practical threshold RSA signatures without a trusted dealer. In *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria*, pages 152–165, 2001.

[dMW06] Michael de Mare and Rebecca N. Wright. Secure set membership using 3Sat. In *Information and Communications Security, 8th International Conference, ICICS 2006, Raleigh, NC, USA*, pages 452–468, 2006.

[DT08] Ivan Damgård and Nikos Triandopoulos. Supporting non-membership proofs with bilinear-map accumulators. *IACR Cryptology ePrint Archive*, 2008:538, 2008. http://eprint.iacr.org/2008/538.

[EEC15] ENEC Estonian National Electoral Committee, EIVC Estonian Internet Voting Committee, and Cybernetica AS. e-hääletamise tarkvara. Technical report, ENEC-EIVC-C, 2015. https://github.com/vvk-ehk/evalimine.

[ElG84] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA*, pages 10–18, 1984.

[Elk10] Michael Elkin. An improved construction of progression-free sets. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA*, pages 886–905, 2010.

[Eur06] European Parliament, Council. Regulation (EC) no 562/2006 of the european parliament and of the council of 15 march 2006 establishing a community code on the rules governing the movement of persons across borders (Schengen Borders Code). http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32006R0562:EN:NOT, March 2006.

[Eve81] Shimon Even. Protocol for signing contracts. In *Advances in Cryptology: A Report on CRYPTO 81, CRYPTO 81, IEEE Workshop on Communications Security, Santa Barbara, California, USA*, pages 148–153, 1981.

[FCoS09] Federal Chancellery of Switzerland. Votation populaire du 17.05.2009 (Votation No 542, Tableau récapitulatif). https://www.admin.ch/ch/f/pore/va/20090517/det542.html, 2009.

[FFS88] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *J. Cryptology*, 1(2):77–94, 1988.

[Fis01] Marc Fischlin. A cost-effective pay-per-multiplication comparison method for millionaires. In *Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA*, pages 457–472, 2001.

[FLZ13] Prastudy Fauzi, Helger Lipmaa, and Bingsheng Zhang. Efficient modular NIZK arguments from shift and product. In *Cryptology and Network Security - 12th International Conference, CANS 2013, Paraty, Brazil*, pages 92–121, 2013.

[FLZ14] Prastudy Fauzi, Helger Lipmaa, and Bingsheng Zhang. Efficient non-interactive zero knowledge arguments for set operations. In *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados*, pages 216–233, 2014.

[FO97] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA*, pages 16–30, 1997.

[FO98] Eiichiro Fujisaki and Tatsuaki Okamoto. A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland*, pages 32–46, 1998.

[Fri] Friedhelm. 2012 facts and figures on Frankfurt airport. http://www.frankfurt-airport.com/content/frankfurt_airport/en/misc/container/facts-and-figures-2011/jcr:content.file/zadafa-2012_e_lowres.pdf.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA*, pages 186–194, 1986.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, Baltimore, Maryland, USA*, pages 416–426, 1990.

[GGM14] Christina Garman, Matthew Green, and Ian Miers. Decentralized anonymous credentials. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA*, 2014.

[GHKR08] Rosario Gennaro, Shai Halevi, Hugo Krawczyk, and Tal Rabin. Threshold RSA for Dynamic and Ad-Hoc Groups. In *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey*, pages 88–107, 2008.

[GJM02] Philippe Golle, Stanislaw Jarecki, and Ilya Mironov. Cryptographic primitives enforcing communication and storage complexity. In *Financial Cryptography, 6th International Conference, FC 2002, Southampton, Bermuda*, pages 120–135, 2002.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the fiat-shamir paradigm. In *44th Symposium on Foundations of Computer Science (FOCS 2003), Cambridge, MA, USA*, pages 102–113, 2003.

[GKZ14] Robert Granger, Thorsten Kleinjung, and Jens Zumbrägel. Breaking '128-bit Secure' Supersingular Binary Curves - (Or How to Solve Discrete Logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$). In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Part II, Santa Barbara, CA, USA*, pages 126–145, 2014.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, Providence, Rhode Island, USA*, pages 291–304, 1985.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[GMSV13] Fuchun Guo, Yi Mu, Willy Susilo, and Vijay Varadharajan. Membership encryption and its applications. In *Information Security and Privacy - 18th Australasian Conference, ACISP 2013, Brisbane, Australia*, pages 219–234, 2013.

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *J. Cryptology*, 7(1):1–32, 1994.

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.

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

[GPS08] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[GR15] Mohit Garg and Jaikumar Radhakrishnan. Set membership with a few bit probes. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA*, pages 776–784, 2015.

[Gro05] Jens Groth. Non-interactive zero-knowledge arguments for voting. In *Applied Cryptography and Network Security, Third International Conference, ACNS 2005, New York, NY, USA*, pages 467–482, 2005.

[Gro10] Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore*, pages 321–340, 2010.

[Gro11] Jens Groth. Efficient zero-knowledge arguments from two-tiered homomorphic commitments. In *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea*, pages 431–448, 2011.

[GS07] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. *IACR Cryptology ePrint Archive*, 2007:155, 2007. http://eprint.iacr.org/2007/155.

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey*, pages 415–432, 2008.

[GS12a] Jens Groth and Amit Sahai. Efficient noninteractive proof systems for bilinear groups. *SIAM J. Comput.*, 41(5):1193–1232, 2012.

[GS12b] Divya Gupta and Amit Sahai. On constant-round concurrent zero-knowledge from a knowledge assumption. *CoRR*, abs/1210.3719, 2012.

[GSNB11] C. C. F. Pereira Geovandro, Marcos A. Simplício, Jr., Michael Naehrig, and Paulo S. L. M. Barreto. A family of implementation-friendly BN elliptic curves. *Journal of Systems and Software*, 84(8):1319–1326, 2011.

[GTH02] Michael T. Goodrich, Roberto Tamassia, and Jasminka Hasic. An efficient dynamic and distributed cryptographic accumulator. In *Information Security, 5th International Conference, ISC 2002 Sao Paulo, Brazil*, pages 372–388, 2002.

[GTH09] Michael T. Goodrich, Roberto Tamassia, and Jasminka Hasic. An efficient dynamic and distributed RSA accumulator. *CoRR*, abs/0905.1307, 2009. http://arxiv.org/abs/0905.1307.

[Han06] Gerhard P. Hancke. Practical attacks on proximity identification systems (short paper). In *2006 IEEE Symposium on Security and Privacy (S&P 2006), Berkeley, California, USA*, pages 328–333, 2006.

[HHJ+06] Jaap-Henk Hoepman, Engelbert Hubbers, Bart Jacobs, Martijn Oostdijk, and Ronny Wichers Schreur. Crossing borders: Security and privacy issues of the european e-passport. In *Advances in Information and Computer Security, First International Workshop on Security, IWSEC 2006, Kyoto, Japan*, pages 152–167, 2006.

[HR07] Martin Hlavác and Tomás Rosa. A note on the relay attacks on e-passports: The case of Czech e-passports. *IACR Cryptology ePrint Archive*, 2007:244, 2007. http://eprint.iacr.org/2007/244.

[HSV06] Florian Hess, Nigel P. Smart, and Frederik Vercauteren. The Eta pairing revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, 2006.

[HZ06] Alexander Herrigel and Jian Zhao. RFID identity theft and countermeasures. *Optical Security and Counterfeit Deterrence Techniques VI*, 6075(1):366–379, 2006.

[ICAO04a] ICAO International Civil Aviation Organization. Machine readable travel documents. Development of a logical data structure — LDS for optional capacity expansion technologies. http://www.icao.int/mrtd/download/technical.cfm, 2004. Version 1.7.

[ICAO04b] ICAO International Civil Aviation Organization. Machine readable travel documents. PKI for machine readable travel documents offering ICC read-only access. http://www.icao.int/mrtd/download/technical.cfm, 2004. Version 1.1.

[ICAO06] ICAO International Civil Aviation Organization. Machine readable travel documents. part 1: Machine readable passport, specifications for electronically enabled passports with biometric identification capabilities. http://www2.icao.int/en/MRTD/Pages/default.aspx, 2006. ICAO Doc 9303.

[ICAO08] ICAO International Civil Aviation Organization. Machine readable travel documents – document 9303. Technical report, ICAO, 2005-2008. http://www.icao.int/Security/mrtd/Pages/Document9303.aspx.

[ICAO13] ICAO International Civil Aviation Organization. Machine readable travel documents – SUPPLEMENT to document 9303. Technical report, ICAO, 2013. http://www.icao.int/Security/mrtd/Pages/Document9303.aspx.

[IIDEA02] International IDEA, International Institute for Democracy and Electoral Assistance. *Voter turnout since 1945 : a global report*. Stockholm, 2002. 3. rev, Updated.

[IP99] Russell Impagliazzo and Ramamohan Paturi. The complexity of k-SAT. *2012 IEEE 27th Conference on Computational Complexity*, 0:237, 1999.

[ISO10a] ISO/IEC 14443:2010. *Identification Cards — Contactless Integrated Circuit(s) Cards — Proximity Cards*. ISO, Geneva, Switzerland, 2010.

[ISO10b] ISO/IEC 18033-3:2010. *Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers*. ISO, Geneva, Switzerland, 2010.

[ITU15] ITU International Telecommunication Union. Child Online Protection. http://www.itu.int/en/cop, 2015.

[JMW05] Ari Juels, David Molnar, and David Wagner. Security and privacy issues in e-passports. In *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 74–88, Washington, DC, USA, 2005. IEEE Computer Society.

[KAF+10] Thorsten Kleinjung, Kazumaro Aoki, Jens Franke, Arjen K. Lenstra, Emmanuel Thomé, Joppe W. Bos, Pierrick Gaudry, Alexander Kruppa, Peter L. Montgomery, Dag Arne Osvik, Herman J. J. te Riele, Andrey Timofeev, and Paul Zimmermann. Factorization of a 768-bit RSA modulus. In *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA*, pages 333–350, 2010.

[KFHS06] Marijana Kosmerlj, Tom Fladsrud, Erik Hjelmås, and Einar Snekkenes. Face recognition issues in a border control environment. In *Advances in Biometrics, International Conference, ICB 2006, Hong Kong, China*, pages 33–39, 2006.

[Kin00] Brian King. Improved methods to perform threshold RSA. In *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan*, pages 359–372, 2000.

[KK05] Gaurav S. Kc and Paul A. Karger. Preventing attacks on machine readable travel documents (MRTDs). *IACR Cryptology ePrint Archive*, 2005:404, 2005. http://eprint.iacr.org/2005/404.

[KZG10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore*, pages 177–194, 2010.

[LAN02] Helger Lipmaa, N. Asokan, and Valtteri Niemi. Secure vickrey auctions without threshold trust. In *Financial Cryptography, 6th International Conference, FC 2002, Southampton, Bermuda*, pages 87–101, 2002.

[Len06] Arjen K. Lenstra. Key lengths. In Hossein Bidgoli, editor, *The Handbook of Information Security*. Wiley, 2006.

[LG07] Dimitrios Lekkas and Dimitris Gritzalis. E-passports as a means towards the first world-wide public key infrastructure. In *Public Key Infrastructure, 4th European PKI Workshop: Theory and Practice, EuroPKI 2007, Palma de Mallorca, Spain*, pages 34–48, 2007.

[Lin01] Yehuda Lindell. Parallel coin-tossing and constant-round secure two-party computation. In *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA*, pages 171–189, 2001.

[Lin03] Yehuda Lindell. Parallel coin-tossing and constant-round secure two-party computation. *J. Cryptology*, 16(3):143–184, 2003.

[Lin15] Yehuda Lindell. An efficient transform from sigma protocols to NIZK with a CRS and non-programmable random oracle. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, Proceedings, Part I*, pages 93–109, 2015.

[Lip03] Helger Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In *Advances in Cryptology - ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan*, pages 398–415, 2003.

[Lip11] Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. *IACR Cryptology ePrint Archive*, 2011:009, 2011. http://eprint.iacr.org/2011/009.

[Lip12a] Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy*, pages 169–189, 2012.

[Lip12b] Helger Lipmaa. Secure accumulators from euclidean rings without trusted setup. In *Applied Cryptography and Network Security - 10th International Conference, ACNS 2012, Singapore*, pages 224–240, 2012.

[Lip14a] Helger Lipmaa. Efficient NIZK arguments via parallel verification of benes networks. In *Security and Cryptography for Networks - 9th International Conference, SCN 2014, Amalfi, Italy*, pages 416–434, 2014.

[Lip14b] Helger Lipmaa. Prover-efficient commit-and-prove zero-knowledge SNARKs. *IACR Cryptology ePrint Archive*, 2014:396, 2014. http://eprint.iacr.org/2014/396.

[Lip16] Helger Lipmaa. Prover-efficient commit-and-prove zero-knowledge SNARKs. In *Progress in Cryptology - AFRICACRYPT 2016 - 8th International Conference on Cryptology in Africa, Fes, Morocco*, pages 185–206, 2016.

[LKLRP07] Yifei Liu, Timo Kasper, Kerstin Lemke-Rust, and Christof Paar. E-passport: Cracking basic access control keys. In *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS, OTM Confederated International Conferences CoopIS, DOA, ODBASE, GADA, and IS 2007, Vilamoura, Portugal, Part II*, pages 1531–1547, 2007.

[LLX07] Jiangtao Li, Ninghui Li, and Rui Xue. Universal accumulators with efficient nonmembership proofs. In *Applied Cryptography and Network Security, 5th International Conference, ACNS 2007, Zhuhai, China*, pages 253–269, 2007.

[LMSF06] Mikko Lehtonen, Florian Michahelles, Thorsten Staake, and Elgar Fleisch. Strengthening the security of machine readable documents by combining RFID and optical memory devices. In *Developing ambient intelligence - proceedings of the first international conference on ambient intelligence developments*, pages 77–92. Springer Paris, 2006.

[Low96] Gavin Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Tools and Algorithms for Construction and Analysis of Systems, Second International Workshop, TACAS '96, Passau, Germany*, pages 147–166, 1996.

[LRSW99] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In *Selected Areas in Cryptography, 6th Annual International Workshop, SAC'99, Kingston, Ontario, Canada*, pages 184–199, 1999.

[LY12] Benoît Libert and Moti Yung. Non-interactive CCA-secure threshold cryptosystems with adaptive security: New framework and constructions. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy*, pages 75–93, 2012.

[LZ12] Helger Lipmaa and Bingsheng Zhang. A more efficient computationally sound non-interactive zero-knowledge shuffle argument. In *Security and Cryptography for Networks - 8th International Conference, SCN 2012, Amalfi, Italy*, pages 477–502, 2012.

[LZJX10] C. H. Li, X. F. Zhang, H. Jin, and W. Xiang. E-passport EAC scheme based on identity-based cryptography. *Inf. Process. Lett.*, 111(1):26–30, 2010.

[Mao98] Wenbo Mao. Guaranteed correct sharing of integer factorization with off-line shareholders. In *Public Key Cryptography, First International Workshop on Practice and Theory in Public Key Cryptography, PKC '98, Pacifico Yokohama, Japan*, pages 60–71, 1998.

[Mat02] Tsutomu Matsumoto. Gummy and conductive silicone rubber fingers. In *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand*, pages 574–576, 2002.

[MN10] Tal Moran and Moni Naor. Split-ballot voting: Everlasting privacy with distributed trust. *ACM Trans. Inf. Syst. Secur.*, 13(2), 2010.

[MPV09] Jean Monnerat, Sylvain Pasini, and Serge Vaudenay. Efficient deniable authentication for signatures. In *Applied Cryptography and Network Security, 7th International Conference, ACNS 2009, Paris-Rocquencourt, France*, pages 272–291, 2009.

[MRK03] Silvio Micali, Michael O. Rabin, and Joe Kilian. Zero-knowledge sets. In *44th Symposium on Foundations of Computer Science (FOCS 2003), Cambridge, MA, USA*, pages 80–91, 2003.

[MVV07] Jean Monnerat, Serge Vaudenay, and Martin Vuagnoux. About Machine-Readable Travel Documents. In *In Proceedings of the International Conference on RFID Security 2007*, Lecture Notes in Computer Science, pages 15–28. Springer, 2007.

[Ngu05] Lan Nguyen. Accumulators from bilinear pairings and applications. In *Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA*, pages 275–292, 2005.

[NIST12] NIST National Institute of Standards and Technology. Recommendation for Key Management – Part 1: General (Revision 3), NIST Special Publication 800-57. Technical report, NIST, July 2012. http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf.

[NIST15a] NIST National Institute of Standards and Technology. FIPS PUB 180-4: Secure hash standard, August 2015. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.

[NIST15b] NIST National Institute of Standards and Technology. FIPS PUB 202: Sha-3 standard: Permutation-based hash and extendable-output functions, August 2015. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf.

[Nit09] Rishab Nithyanand. A survey on the evolution of cryptographic protocols in epassports. *IACR Cryptology ePrint Archive*, 2009:200, 2009. http://eprint.iacr.org/2009/200.

[NNPC10] Ahmet Erhan Nergiz, Mehmet Ercan Nergiz, Thomas Pedersen, and Chris Clifton. Practical and secure integer comparison and interval check. In *Proceedings of the 2010 IEEE Second International Conference on Social Computing, SocialCom / IEEE International Conference on Privacy, Security, Risk and Trust, PASSAT 2010, Minneapolis, Minnesota, USA*, pages 791–799, 2010.

[ODIHR12] ODIHR Office for Democratic Institutions and Human Rights. OSCE/ODIHR Election Assessment Mission Report, Swiss Confederation, Federal Assembly

Elections 23 October 2011. Technical report, OSCE/ODIHR, January 2012. http://www.osce.org/odihr/elections/Switzerland/83755.

[OOK90] Kazuo Ohta, Tatsuaki Okamoto, and Kenji Koyama. Membership authentication for hierarchical multigroups using the extended fiat-shamir scheme. In *Advances in Cryptology - EUROCRYPT '90, Workshop on the Theory and Application of Cryptographic Techniques, Aarhus, Denmark*, pages 446–457, 1990.

[ORS04] Rafail Ostrovsky, Charles Rackoff, and Adam Smith. Efficient consistency proofs for generalized queries on a committed database. In *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland*, pages 1041–1053, 2004.

[Osm15] Murat Osmanoglu. *Graded Cryptographic Primitives*. PhD thesis, University of Connecticut, 2015.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic*, pages 223–238, 1999.

[PBD00] DongGook Park, Colin Boyd, and Ed Dawson. Classification of authentication protocols: A practical approach. In *Information Security, Third International Workshop, ISW 2000, Wollongong, NSW, Australia*, pages 194–208, 2000.

[Ped91] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA*, pages 129–140, 1991.

[PPW08a] Vijayakrishnan Pasupathinathan, Josef Pieprzyk, and Huaxiong Wang. Formal security analysis of australian e-passport implementation. In Ljiljana Brankovic and Mirka Miller, editors, *Sixth Australasian Information Security Conference (AISC 2008)*, volume 81 of *CRPIT*, pages 75–82, Wollongong, NSW, Australia, 2008. ACS.

[PPW08b] Vijayakrishnan Pasupathinathan, Josef Pieprzyk, and Huaxiong Wang. An online secure e-passport protocol. In *Information Security Practice and Experience, 4th International Conference, ISPEC 2008, Sydney, Australia*, pages 14–28, 2008.

[PPW08c] Vijayakrishnan Pasupathinathan, Josef Pieprzyk, and Huaxiong Wang. Security analysis of Australian and E.U. e-passport implementation. *Journal of Research and Practice in Information Technology*, 40(3):187–205, 2008.

[PS96] David Pointcheval and Jacques Stern. Security proofs for signature schemes. In *Advances in Cryptology - EUROCRYPT '96, International Conference on the*

Theory and Application of Cryptographic Techniques, Saragossa, Spain, pages 387–398, 1996.

[PS15] David Pointcheval and Olivier Sanders. Short randomizable signatures. *IACR Cryptology ePrint Archive*, 2015:525, 2015. http://eprint.iacr.org/2015/525.

[PTT08] Charalampos Papamanthou, Roberto Tamassia, and Nikos Triandopoulos. Authenticated hash tables. In *Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA*, pages 437–448, 2008.

[RKP09] Alfredo Rial, Markulf Kohlweiss, and Bart Preneel. Universally composable adaptive priced oblivious transfer. In *Pairing-Based Cryptography - Pairing 2009, Third International Conference, Palo Alto, CA, USA*, pages 231–247, 2009.

[RS86] Michael O. Rabin and Jeffery O. Shallit. Randomized algorithms in number theory. *Communications on Pure and Applied Mathematics*, 39(S1):S239–S256, 1986.

[RS04] Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India*, pages 371–388, 2004.

[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.

[RSV02] Jaikumar Radhakrishnan, Pranab Sen, and Srinivasan Venkatesh. The quantum complexity of set membership. *Algorithmica*, 34(4):462–479, 2002.

[San99] Tomas Sander. Efficient accumulators without trapdoor extended abstracts. In *Information and Communication Security, Second International Conference, ICICS'99, Sydney, Australia*, pages 252–262, 1999.

[San11] Tom Sanders. On Roth's theorem on progressions. *Annals of Mathematics. Second Series*, 174(1):619–636, 2011.

[Sce09] Antoine Scemama. *A cryptanalysis of the 2R cryptosystem and an improved commitment range proof*. PhD thesis, Goethe University Frankfurt am Main, 2009.

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3):161–174, 1991.

[Sch01] Berry Schoenmakers. Some efficient zeroknowledge proof techniques. Slides presented at the *International Workshop on Cryptographic Protocols*, March 2001. Monte Verita, Switzerland.

[Sch05] Berry Schoenmakers. Interval proofs revisited. Slides presented at the *International Workshop on Frontiers in Electronic Elections*, September 2005. Milan, Italy.

[SCP94] Alfredo De Santis, Giovanni Di Crescenzo, and Giuseppe Persiano. The knowledge complexity of quadratic residuosity languages. *Theor. Comput. Sci.*, 132(2):291–317, 1994.

[SFD+14] Drew Springall, Travis Finkenauer, Zakir Durumeric, Jason Kitcat, Harri Hursti, Margaret MacAlpine, and J. Alex Halderman. Security analysis of the estonian internet voting system. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA*, pages 703–715, 2014.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA*, pages 47–53, 1984.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany*, pages 256–266, 1997.

[Sho00] Victor Shoup. Practical threshold signatures. In *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium*, pages 207–220, 2000.

[SMI91] Chaosheng Shu, Tsutomu Matsumoto, and Hideki Imai. A multi-purpose proof system - for identity and membership proofs. In *Advances in Cryptology - ASIACRYPT '91, International Conference on the Theory and Applications of Cryptology, Fujiyoshida, Japan*, pages 397–411, 1991.

[SMP87] Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Non-interactive zero-knowledge proof systems. In *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA*, pages 52–72, 1987.

[SRA81] Adi Shamir, Ronald L Rivest, and Leonard M Adleman. *Mental poker*. Springer, 1981.

[TS06] Isamu Teranishi and Kazue Sako. *K*-times anonymous authentication with a constant proving cost. In *Public Key Cryptography - PKC 2006, 9th International Conference on Theory and Practice of Public-Key Cryptography, New York, NY, USA*, pages 525–542, 2006.

[TV06] Terence Tao and Van H. Vu. *Additive Combinatorics*, volume 105 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, September 2006.

[TX03] Gene Tsudik and Shouhuai Xu. Accumulating composites and improved group signing. In *Advances in Cryptology - ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan*, pages 269–286, 2003.

[Vau06] Serge Vaudenay. *A Classical Introduction to Cryptography*. Springer, 2006. http://www.vaudenay.ch/crypto/.

[Vau07] Serge Vaudenay. E-passport threats. *IEEE Security & Privacy*, 5(6):61–64, 2007.

[Vol99] Heribert Vollmer. *Introduction to Circuit Complexity - A Uniform Approach*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 1999.

[vTJ11] Henk C. A. van Tilborg and Sushil Jajodia, editors. *Encyclopedia of Cryptography and Security, 2nd Ed.* Springer, 2011.

[VV07] Serge Vaudenay and Martin Vuagnoux. About Machine-Readable Travel Documents. *Journal of Physics: Conference Series*, 77(1):012006, 2007. http://iopscience.iop.org/1742-6596/77/1/012006/pdf/jpconf7_77_012006.pdf.

[WHLD14] Genqiang Wu, Yeping He, Yi Lu, and Liping Ding. Efficient interval check in the presence of malicious adversaries. *IACR Cryptology ePrint Archive*, 2014:690, 2014. http://eprint.iacr.org/2014/690.

[WWP07] Peishun Wang, Huaxiong Wang, and Josef Pieprzyk. A new dynamic accumulator for batch updates. In *Information and Communications Security, 9th International Conference, ICICS 2007, Zhengzhou, China*, pages 98–112, 2007.

[YHM+09] Tsz Hon Yuen, Qiong Huang, Yi Mu, Willy Susilo, Duncan S. Wong, and Guomin Yang. Efficient non-interactive range proof. In *Computing and Combinatorics, 15th Annual International Conference, COCOON 2009, Niagara Falls, NY, USA*, pages 138–147, 2009.

List of Protocols

Index

Rafik CHAABOUNI

Chemin des Sarments 2B,
1222 Vésenaz, Switzerland
(+41) 78 850 2555
Rafik@Chaabouni.ch

View my profile on LinkedIn

Strengths

- Cyber Security Expert
- Privacy specialist:
  - set membership
  - biometric passports
  - revocation
- Multicultural (lived in 6 countries)

Education

2017 **EPFL - École Polytechnique Fédérale de Lausanne**
Doctor of Science in Cryptography and Security
- Focus on Trust and Privacy, with an Academic visit (5 years) at University of Tartu, Estonia
- Supervisor: Professor Serge Vaudenay
- Co-Supervisor: Professor Helger Lipmaa

2007 **EPFL - École Polytechnique Fédérale de Lausanne**
Master of Science in Communication Systems
- Specialization in Information and Communication Security, with a Minor in Management of Technology and Entrepreneurship
- Academic exchange year at CMU - Carnegie Mellon University, USA (awarded only to top 10 students among 200)

2002 **Swiss Scientific Federal Maturité** (equivalent high school diploma)

Core Experience

2007 - 2017 **Ph.D. Thesis,** Title: "Enhancing Privacy Protection"
- Sub-title: "Set Membership, Range Proofs, and the Extended Access Control"
- Improvement of cryptographic primitives for privacy: Set Membership Proofs, Interactive, and Non-Interactive Range Proofs
- Improvement of standards for Machine Readable Travel Documents (MRTD)
- Creation of a solution for Terminal Revocation in the Extended Access Control (EAC)

2011 - 2016 **Cryptography Research Group, University of Tartu, Estonia**
- International Visiting Ph.D. assistant
- Project Manager, Team Manager, External Reviewer, Master thesis Opponent, Lecturer, and Teaching assistant
- Searching, applying, and obtaining grants (> 32'000 €)
- Initiator and Manager of the Crypto/Security Reading Group
- Program committee member of the 19th Nordic Conference on Secure IT Systems (NordSec 2014)

2007 - 2011 **Security and Cryptography Laboratory, LASEC, EPFL**
- Doctoral assistant, Project Manager, External Reviewer, Teaching assistant
- Searching, applying, and obtaining grants (> 200'000 CHF)
- Administrator for the Linux network of the lab (gateway, web server, data server, and integration of Kerberos)

2007 **IBM, Zürich Research Lab** (6-month)
- Intern, master thesis project
- Supervisor: Dr. Jan Camenisch
- Study of past Zero-Knowledge Set Membership and Range Proof
- Improvement by the creation of more efficient protocols that resulted in a *patent,* and led to a *publication* during my Ph.D.

Patents

2012 *Methods for efficient certificate revocation list compression* (WO Patent)
A. Mashatan, I. Aad, R. Chaabouni, P. V. Niemi, S. Vaudenay

2011 *Set membership proofs in data processing systems* (US Patent)
J. Camenisch, R. Chaabouni, A. Shelat

Additional Experience

2009 - 2010 **Nokia Research Center, Lausanne**
- Researcher: Study and Development of Revocation in Mobile Environment, that resulted in a *patent*

2006 **Caprices Festival, Crans-Montana**
- Consultant: Disclosure of Security Vulnerability (SQL injection)

2005 **International Telecommunication Union (ITU), Geneva** (4-month)
- Intern
- Study and recommendations on minutes trading and minutes exchange for IP telephony
- Analysis and recommendations for developing countries in the subject of 4G
- Analysis of actual state of Arabic Domain Names and recommendations

Grants

(2007 - 2016)

- European Commission Seventh Framework Programme (FP7/2007-2013)
- Swiss National Science Foundation (SNF 200021-124575)
- European Network of Excellence in Cryptology II (ECRYPT II, ICT-2007-216676)
- Estonian Science Foundation (ESF #9303)
- European Regional Development Fund
- DoRa (EU Doctoral Studies and Internationalization Programme)
- IUT2-1 (Institutsionaalsed Uurimistoetused)
- Estonian Center of Excellence in Computer Science (EXCS)

Publications

[C13] *Solving Terminal Revocation in EAC by Augmenting Terminal Authentication*.
R. Chaabouni - BIOSIG 2013

[CLZ12] *A non-interactive range proof with constant communication*.
R. Chaabouni, H. Lipmaa, and B. Zhand - Financial Cryptography 2012

[CLS10] *Additive combinatorics and discrete logarithm based range protocols*.
R. Chaabouni, H. Lipmaa, and A. Shelat - ACISP 2010

[CV09] *The extended access control for machine readable travel documents*.
R. Chaabouni and S. Vaudenay - BIOSIG 2009

[CCS08] *Efficient protocols for set membership and range proofs.*
J. Camenisch, R. Chaabouni, and A. Shelat - ASIACRYPT 2008 - (145 citations)

Full list at: Infoscience.epfl.ch

Invited Speaker

1st (2010) **and 2nd** (2012) **Arab Forum of the Arab ICT Organization:**
"e-transactions Security & the Public Key Infrastructure (PKI)"
2010: Presentation of [CV09] for the Arab ministries of communication technologies
2012: Presentation of [C12a] and [C12b]

Selected Projects

2014 Drones InSecurity (Manuel Kramer and Martin Schmeier - UT)
Data Storage Leakage and Manipulation (Siddharth Prakash Rao – UT)
From simple keylogger to advanced logging (Dmitri Gabbasov - UT)
2010 TCHo implementation in hardware (Vincent Bindschaedler - EPFL)
2009 Identity Based Encryption (Alexandre Duc - EPFL)
Rethinking the PKI Trust Model (Sabrina Perez - EPFL)
Mistrusting Sellers and Malicious Buyers (Teodora Kostic - EPFL)

Languages

English	fluent
French	bilingual (mother tongue)
Arabic	bilingual (mother tongue)
German	medium (4 years at school)
Estonian	A2 level

Technical Skills

(among many others)

Security Protocols: WEP, RC4, WPA, Kerberos, DH, RSA, SSH, TLS, X.509
Security Tools: Nmap, Aircrack, Kismet, Wireshark, tcpdump, Ophcrack, John the Ripper
Programming: Java, C, Python, Perl, MATLAB, MySQL, PostgreSQL, PARI/GP
Web Development: HTML, PHP, CSS, JavaScript, Jahia, WordPress, Gallery
OS: Windows, OSX, Solaris, Linux (Kali, Debian, Raspbian, Knoppix, Ubuntu, Gentoo)
Office Tools: MS Office (Word, Excel, PowerPoint), iWork (Pages, Numbers, Keynote), KDE, OpenOffice, LaTeX, XYpic, Ti*k*Z, Git, SVN

Extra-Curricular

Social Activities:

Vice-President of PolyCube 2008 - 2010
- Promote the Rubik's Cube at EPFL
- Organized the 1st International Competition of Switzerland (Swiss Open 2009), with 30 international competitors

Logisitics Manager of PolyJapan 2008 - 2010
- Promote the Japanese culture at EPFL
- Organized 1st and 2nd Convention: Japan Impact 2009 and 2010 (2'000 visitors, 2 days, 6'000 m^2)

Sports: Scuba diving (PADI: Advanced Open Water Diver), climbing (Class 5, top rope, and lead), and snowboarding/skiing

Personal Details

Marital status: Single (no kids)
Date of birth: 31 August 1984 (age: 32)
Nationality: Swiss and Tunisian (discharged from military service)