Cryptography in Privacy-preserving Applications

Tsang Pak Kong

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy
in
Information Engineering

June 2005

Cryptography in Privacy-preserving Applications

submitted by

Tsang Pak Kong

for the degree of Master of Philosophy
at the Chinese University of Hong Kong

Abstract

We propose two linkable ring signature schemes for privacy-preserving applications. They are *short linkable ring signature scheme* and *separable linkable threshold ring signature scheme*. The *short linkable ring signature scheme* is the first linkable ring signature scheme that produces signatures of size independent of group size. This makes the scheme scalable and very practical to be deployed in a large scale. The *separable linkable threshold ring signature scheme* is the first of its kind to support separability and efficient thresholding. Separability allows users of a scheme to be heterogenous from security parameters to cryptographic primitives and therefore is a favorable property in ad hoc networks.

We discuss and rigorously define notions of security and functionality that have never been considered in the literature, such as *accusatory linking* and *non-slanderability*. Accusatory linking identifies a cheating signer and hence discourages cheating. Accusatorily linkable ring signatures therefore find new applications. Non-slanderability ensures honest users cannot be slandered on. It is a vital property that should be possessed by all linkable ring signature schemes. We formulate a security model for linkable (threshold) ring signature schemes and prove the security of our two proposed constructions under

the model, under some well-known mathematical assumptions and the *Link Decisional RSA* (LD-RSA) Assumption we formulate.

We investigate three challenging privacy-preserving applications. They are *offline anonymous electronic cash*, *electronic voting* and *anonymous attestation*. They all face a thorny and contradicting difficulty – on one hand users want their privacy to be maintained, on the other the authority wants authentication for eligibility. We show how to use our proposed schemes to implement all the three of them.

摘要

我們提出兩個可用於隱私維護應用鏈結性環簽名 (Linkable Ring Signature) 方案。它們是短鏈結性環簽名 (Short Linkable Ring Signature) 方案和可分離鏈結性門限環簽名 (Separable Linkable Threshold Ring Signature) 方案。短鏈結性環簽名方案爲首個簽名大小與群大小無關的鏈結性環簽名方案，使此方案能有效地大規模落實應用。可分離鏈結性門限環簽名方案是同類中首個支持可分離性和有效率門限化，可分離性容許的使用者擁有相異安全參數 (security parameters) 甚至密碼基礎 (cryptographic primitives)，是個於在無基礎架構網路 (Ad Hoc Networks) 有利的特點。

我們討論並精確地定義文獻上未被提及的安全和功能概念，例如具指控性的鏈結性 (Accusatory Linking) 和不能誹謗性 (Non-slanderability)。具指控性的鏈結性能識別作弊的簽名者，因而能被用於新的應用上。不能誹謗性保證誠實使用者不能被誹謗，是所有鏈結性環簽名方案都應具備的重要屬性。我們爲鏈結性(門限)環簽名制定安全模型，以及證明，基於一些常見的數學前設和我們制定的 Link Decisional RSA 前設下，我們提出的兩個建構在此模型下是安全的。

我們探討三個具挑戰性的隱私維護應用。它們是離線匿名電子金錢 (Offline Anonymous Electronic Cash)，電子投票 (Electronic Voting) 和匿名證名 (Anonymous Attestation)。它們同時面對一個棘手而矛盾的難題：一方面使用者希望他們的隱私受到維護，另一方面管理機構希望鑑定使用者的合法性。我們展示如何使用我們提出的方案來實行以上三個應用。

Acknowledgment

First of all, I am profoundly grateful to Victor K. Wei, my advisor. On top of being a senior professor and an expert in the field, Victor is also a devoted advisor far beyond the ordinary. I must say that his dedication for pursuing the best in every single detail is just amazing. He is always acute and to-the-point when identifying a research problem at hand. Victor guided me into the fascinating field of Cryptography and the basis for this thesis lies in the joint work with him (and some others). Many of the results reported herein are the fruits of lively discussions with him. I am so influenced not only by his research methodology but also the respectful attitude towards one's career. I am not going to forget all this in the rest of my life.

I am also thankful for Joseph Liu and Duncan Wong. They have been like my mentors during the two years of my master study. They offered a lot of help to a novice in research like me in getting onto the track. The time I spent with them, no matter it was on discussing research matters or any other topics, was always enjoyable.

Next I wish to thank my colleagues. They are Allen Au, Patrick Chan, Tony Chan, Sebastian Fleissner, Karyin Fung, Robert Leung, and Tsz Hon Yuen. Life in the Information Security Lab would have been much less colorful without them. It was so good to have someone to talk to during the time I was depressed, and these guys were especially good listeners because they did understand where the difficulties laid. We also had gatherings from time to time and a game of Winning Eleven was already so much fun. Days of joy, days of sadness, I certainly could not have done this without my friends.

Last, but certainly not least, I am grateful to my family for their continuous support in every single way. Their unreserved confidence in me enabled me to choose the way I want to go.

Contents

Abstract ii

Acknowledgement iv

1 Introduction 1

1.1 Privacy 1

1.2 Cryptography 5

1.2.1 History of Cryptography 5

1.2.2 Cryptography Today 6

1.2.3 Cryptography For Privacy 7

1.3 Thesis Organization 8

2 Background 10

2.1 Notations 10

2.2 Complexity Theory 11

2.2.1 Order Notation 11

2.2.2 Algorithms and Protocols 11

2.2.3 Relations and Languages 13

2.3 Algebra and Number Theory 14

2.3.1 Groups 14

2.3.2 Intractable Problems 16

2.4 Cryptographic Primitives 18

2.4.1 Public-Key Encryption 18
2.4.2 Identification Protocols 21
2.4.3 Digital Signatures 22
2.4.4 Hash Functions 24
2.4.5 Zero-Knowledge Proof of Knowledge 26
2.4.6 Accumulators 32
2.4.7 Public Key Infrastructure 34
2.5 Zero Knowledge Proof of Knowledge Protocols in Groups of Unknown Order 36
2.5.1 The Algebraic Setting 36
2.5.2 Proving the Knowledge of Several Discrete Logarithms . 37
2.5.3 Proving the Knowledge of a Representation 38
2.5.4 Proving the Knowledge of d Out of n Equalities of Discrete Logarithms 39
2.6 Conclusion 42

3 Related Works **43**
3.1 Introduction 43
3.2 Group-Oriented Signatures without Spontaneity and/or Anonymity 44
3.3 SAG Signatures 46
3.4 Conclusion 49

4 Linkable Ring Signatures **50**
4.1 Introduction 50
4.2 New Notions 52
4.2.1 Accusatory Linking 52
4.2.2 Non-slanderability 53
4.2.3 Linkability in Threshold Ring Signatures 54
4.2.4 Event-Oriented Linking 55
4.3 Security Model 56

4.3.1 Syntax 56
4.3.2 Notions of Security 58
4.4 Conclusion 63

5 Short Linkable Ring Signatures **64**
5.1 Introduction 64
5.2 The Construction 65
5.3 Security Analysis 68
5.3.1 Security Theorems 68
5.3.2 Proofs 68
5.4 Discussion 70
5.5 Conclusion 71

6 Separable Linkable Threshold Ring Signatures **72**
6.1 Introduction 72
6.2 The Construction 74
6.3 Security Analysis 76
6.3.1 Security Theorems 76
6.3.2 Proofs 77
6.4 Discussion 79
6.5 Conclusion 80

7 Applications **82**
7.1 Offline Anonymous Electronic Cash 83
7.1.1 Introduction 83
7.1.2 Construction 84
7.2 Electronic Voting 85
7.2.1 Introduction 85
7.2.2 Construction 87
7.2.3 Discussions 88

7.3 Anonymous Attestation 89
7.3.1 Introduction 89
7.3.2 Construction 90
7.4 Conclusion 91

8 Conclusion **92**

A Paper Derivation **94**

Bibliography **95**

Chapter 1

Introduction

In this chapter, we first take a close look at the word "privacy". Specifically, we discuss what it means from a sociology point of view as well as a scientific classification, and its value and importance to different people. We then introduce Cryptography by going through its historical development and studying its significance nowadays. Finally we argue that Cryptography is a key to realize the attainment of various stringent and often contradicting goals, such as preserving the privacy of the users, of computer protocols.

1.1 Privacy

How much is privacy worth? Expect different answers when you ask different people. To an individual privacy is one of the inherent freedoms of a free society. It is the right of people to keep something secret and not known to anyone else. In this sense, privacy is just priceless. But privacy does have a price sometimes. On one hand, maintaining privacy does not come free of charge, people often need to pay an extra cost for hiding a piece of information that would otherwise have been known by others. In this case, the price of privacy is that extra cost paid or willing to be paid. On the other hand, there are people willing to pay in order to unveil secrets others have been striving hard to keep – think of how much magazines pay for candid pictures

of celebrities taken by paparazzi, and think of how desperate companies try to collect information about spending behavior of their loyal customers. Yet another way to value privacy is to look at the huge amount of money spent every year on lawyers to argue the privacy laws.

History has also shown us that we care very much about our privacy. In 1361, the Justices of the Peace Act In England provided for the arrest of peeping toms and eavesdroppers. In 1776, the Swedish Parliament enacted the Access to Public Records Act that required that all government-held information be used for legitimate purposes.The 1948 Universal Declaration of Human Rights specifically protects territorial and communications privacy. Article 12 states, "No one should be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks on his honour or reputation. Everyone has the right to the protection of the law against such interferences or attacks." In 1995 and 1997, the European Union enacted two directives to ensure consistent levels of protection for its citizens.The directives set a common baseline level of privacy. The 1995 Data Protection Directive set a benchmark for national laws for processing personal information in electronic and manual files, and the 1997 Telecommunications Directive established specific protections covering telephone, digital television, mobile networks, and other telecommunications. In addition, the U.S. Constitution and subsequent laws have given us a right to be left alone.

So what is Privacy? Traditional definitions of privacy, which are often influenced by the "right of the individual to be let alone" [98], separate a person or their actions from a group of persons [44]:

> ''Privacy in our common sense is strongly connected with the idea that there are some things another person should not be able to see or know.''

Privacy may also be defined as the right to determine the amount of personal

information which should be available to others [99]:

> ''Privacy is the claim of individuals, group, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others.''

Privacy is both an intrinsic value as well as an instrumental value serving other goals (e.g., generation of knowledge, profit). Besides these theoretical considerations, privacy also serves pragmatic purposes when it is included in the design of software systems, e.g. resulting in higher acceptance by users.

A more fine-grained description of privacy was given by Flinn and Maurer [48] who identified six levels of anonymity, ranging from the unequivocal assignment of data to a person to the complete disengagement of data from the person. The different levels are as follows:

- *Super-identification.* The user's identity is authenticated by means based on the environment of the user adaptive system. This guarantees that no component of the user adaptive system can counterfeit the identity of the respective user. The assignment of the data needed for authentication to the user or to the components is delegated to an administrative entity outside the system architecture. Examples of this kind of identification and authentication are the X.509 standard [2].

- *Identification.* The user identifies himself and demonstrates knowledge of a secret which is then compared by the system to a stored value. The system is responsible for the confirmation of the user's identity. As an example, this mechanism is often implemented in current operation systems.

- *Latent identification (controlled pseudonyms).* The user identifies himself to the system and adopts one of the defined pseudonyms. Subsequently, he is able to act without revealing his identity to particular components

of the system while acting under a pseudonym. The pseudonym can be revealed under defined circumstances in order to ascertain the identity of the user. For example, this procedure is widely used in box number advertisements.

- *Pseudonymous identification (uncontrolled pseudonyms).* When using the system for the first time, the user decides on a unique pseudonym and a secret which he will also use for following sessions. The system is unable to ascertain the identity of the user, therefore it is also unable to link the pseudonym to the user's identity. This method is used in most Web-based services. It is also used in anonymous remailers which allow email exchange by means of uncontrolled unique pseudonyms.

- *Anonymous identification.* This user gains access to the system by providing a secret without disclosing his identity. The system is unable to distinguish between users which have knowledge about the same secret. The users of the same secret constitute an anonymity set. For instance, a bank account might be managed as a numbered account where clients only have to provide a password to get access.

- *Anonymity.* The user neither identifies nor authenticates himself to the system. The system is unable to distinguish among the users or to differentiate between users. Anonymity is given in most real life situations but not in the World-wide Web, where electronic trails on several layers make it possible to link the current user and his system interactions with additional information to the point of revealing his identity.

1.2 Cryptography

1.2.1 History of Cryptography

The earliest known use of cryptography is found in non-standard hieroglyphs carved into monuments from Egypt's Old Kingdom some 4500 years ago. The Romans are famous for the Caesar cipher and its variations.

The invention of the frequency analysis technique for breaking monoalphabetic substitution ciphers was sometime around AD 1000. It was the most fundamental cryptanalytic advance until WWII. Essentially all ciphers remained vulnerable to this cryptanalytic technique until the invention of the polyalphabetic cipher.

Mathematical methods proliferated in the time leading up to World War II, for example, the application of statistical techniques to cryptanalysis and cipher development and the break into the German Army's Enigma.

The mid-1970s there were two major advances. First was the publication of the draft Data Encryption Standard [80, 81] in the U.S. Federal Register in 1975. It was later adopted and published as a FIPS Publication. The second development, in 1976, was perhaps even more important, for it fundamentally changed the way crypto systems might work. This was the publication of [42] by Diffie and Hellman. It introduced a radically new method of distributing cryptographic keys, which went far toward solving one of the fundamental problems of cryptography, key distribution, and has become known as Diffie-Hellman key exchange. The article also stimulated the almost immediate public development of a new class of enciphering algorithms, the asymmetric key algorithms.

1.2.2 Cryptography Today

Cryptography is traditionally the study of means of converting information from its comprehensible form into an incomprehensible format, making it unreadable without secret knowledge. Today, it is a cross-disciplinary science involving Complexity Theory, Algebra and even Engineering and many others. It serves a core to achieve information security such as confidentiality, authentication, integrity, non-repudiation privacy and etc. Confidentiality keeps information secret to all but the authorized. Authentication ascertains the origin of a message. Integrity protects a message from unauthorized modification. Non-repudiation prevents a sender from denying that he has sent a message. Anonymity aims at hiding the identity of a user.

Our society has entered an era where commerce activities, business transactions and government services have been, and more and more of them will be, conducted and offered over open computer and communications networks such as the Internet. Doing things online has a great advantage of an always-on availability to people in any corner of the world. Here are a few examples of things that have been, or will be done online: Banking, bill payment, home shopping, stock trading, auctions, gambling, fair signing of contracts, time-stamping, voting, ticket booking, interactive games, digital libraries,

From e-mail to cellular communications, from secure Web access to digital cash, cryptography is an essential part of today's information systems. Cryptography helps provide accountability, fairness, accuracy, and confidentiality. It can prevent fraud in electronic commerce and assure the validity of financial transactions. It can prove your identity or protect your anonymity. It can keep vandals from altering your Web page and prevent industrial competitors from reading your confidential documents. And in the future, as commerce and communications continue to move to computer networks, cryptography will become more and more vital.

1.2.3 Cryptography For Privacy

The rapid advancement of technology witnessed many commercial applications launched to provide services in the Internet. The growth of the worldwide Internet user base and with Internet based transactions is reaching an ever-increasing value, it makes sense for the parties involved to secure the Internet. Haphazard handling of financial and personal information can lead to the Internet being constantly associated with fraud and privacy abuses instead of being a viable commerce medium.

Security and privacy have long been important issues forming the basis of numerous democracies around the world. In the digital age, securing personal information and ensuring privacy pose to be issues of paramount concern. Many studies have suggested that a majority of consumers are concerned about when, what and how their personal information is being collected, how this information is being used and whether it is being protected. They want to know whether the information is being sold or shared with others, and if so with whom and for what purposes. They also want to have control over their privacy in today's digital age where strides in telecommunication, storage and software technologies have made monitoring a person's activities effortless.

It is no exaggeration to say that cryptography is the savior of the digital world. Cryptography allows people to carry over the confidence found in the physical world to the electronic world, thus allowing people to do business electronically without worries of deceit and deception. As mentioned in the earlier section, cryptography provides the necessary tools such as encryption, authentication, anonymity and so forth for protocol developers. When properly used, cryptography helps combat various security crimes.

User privacy is, however, often sacrificed for other security concerns. In most situations, user privacy is never a prime concern of protocol developers such as large corporations or the government. Sometimes companies even do

the other way round – they try every possible means to know other customers better. While technology for confidentiality, non-repudiation and etc have been promisingly achieved by cryptography, technology for anonymity has been left behind (possibly due to its hardness in nature, or less attention was paid among the research community, which could be more likely).

1.3 Thesis Organization

In this chapter, we have discussed what privacy is, its value and its importance to different people. We have also introduced Cryptography by going through its historical development and studying its significance nowadays. Finally we have argued that Cryptography is a key to realize the attainment of various stringent and always contradicting goals, such as preserving the privacy of the users, of computer protocols.

Chapter 2 provides the necessary background and foundations of Cryptography that will be used in the subsequent chapters. We first give an introduction to the topics of complexity theories, algebra, number theory. We then proceed to review various cryptographic primitives including encryption, digital signatures, etc. Finally we elaborate on zero-knowledge proof of knowledge protocols.

In Chapter 3, we survey the literature on works related to our thesis. They serve as a good tutorial on various security goals and notions, current state-of-art technology and similarities and differences among schemes. We hope that after reading this chapter, the readers can better understand the incentives that have driven the writing of this thesis, and at the same time better evaluate the contribution of this thesis.

In Chapter 4, we investigate in depth one extension of ring signatures, namely the linkable ring signatures. We first give an introduction of linkable ring signatures and the significance of linkability in ring signatures. Then we

introduce notions novel to linkable ring signatures, together with discussion on their importance and impact. Finally we give a fully developed security model that captures the security requirements of linkable ring signature schemes under various possible adversarial attacks.

In Chapter 5, we propose the first short linkable ring signature scheme construction. By short we mean the signature size is independent of the size of the member group a signature is signed on behalf of. Being short enables linkable ring signatures to be scalable and deployed in large-scale scenarios We propose a new mathematical assumption and then reduce the security of our construction to it plus several well-known mathematical assumptions.

In Chapter 6, we propose the first separable linkable threshold ring signature scheme. Separability is the key for a scheme to be practically deployed in ad hoc environment in which machines are highly heterogenous. Our proposed scheme also supports thresholding efficiently in the sense of computational and storage/communication complexities. We reduce the security of our scheme to well-known mathematical assumptions.

Chapter 7 discusses real-life examples when Cryptography is applied to achieve the stringent and sometimes contradictory requirements of various applications. The three applications we are going to look at are: *Offline Anonymous Electronic Cash*, *E-Voting* and *Anonymous Attestation*.

In Chapter 8, we conclude the thesis.

In Appendix A, we list the papers derived from this thesis.

Chapter 2

Background

Our goal in this chapter is to provide the necessary background and foundations of cryptography that will be used in the subsequent chapters. We first give an introduction to the topics of complexity theories, algebra, number theory. We then proceed to review various cryptographic primitives including encryption, digital signatures, etc. Finally we elaborate on zero-knowledge proof of knowledge protocols.

2.1 Notations

In this thesis, we denote by $\mathbb{N}$ the set of positive integers, by $\mathbb{Z}$ the set of integers, and by $\mathbb{R}$ the set of real numbers. We denote by $[a, b]$ the integers x satisfying $a \leq x \leq b$, by $\lfloor x \rfloor$ the largest integer less than or equal to x, and by $\lceil x \rceil$ the smallest integer greater than or equal to x. $|s|$ means the number of elements in s if s is a finite set, or the length of s if s is a string, or the bit-length/size of s if s is an integer. 1^k represents the string of k ones. When we write $x \in_R X$, we mean x is chosen from the finite set X uniformly at random. If S is a set, $\wp(S)$ denotes the power set of S (i.e. the set of all the subsets of S) while $\wp_d(S)$ denotes the set of all the subsets of S with d elements.

2.2 Complexity Theory

2.2.1 Order Notation

The following is useful when describing the asymptotic behaviors of functions.

Definition 1 (Order Notation.) *$f(n) = O(g(n))$ if there exists a positive constant c and a positive integer n_0 such that $0 \leq f(0) \leq cg(n)$ for all $n \geq n_0$. $f(n) = \Omega(g(n))$ if there exists a positive constant c and a positive integer n_0 such that $0 \leq cg(n) \leq f(n)$ for all $n \geq n_0$. $f(n) = \Theta(g(n))$ if there exists positive constant c_1 and c_2, and a positive integer n_0 such that $c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0$. $f(n) = o(g(n))$ if for any positive constant $c > 0$ there exists a constant $n_0 > 0$ such that $0 \leq f(n) < cg(n)$ for all $n \geq n_0$.*

Definition 2 (Negligibility.) *A negligible function, denoted by $\nu(\lambda)$, is a function $f(\lambda)$ such that for all polynomials $p(\lambda)$, $\nu(\lambda) < 1/p(\lambda)$ holds for all sufficiently large λ. A function is non-negligible if it is not negligible.*

Sometimes we say a probability is *overwhelming* to mean that it is negligibly less than 1.

2.2.2 Algorithms and Protocols

We model algorithms using Turing machines. A deterministic Turing machine is a finite state machine having an infinite read-write tape and the state transitions are completely determined by the input. In a probabilistic Turing machine, the state transitions are determined by the input and the output of coin tosses.

Definition 3 (Algorithm.) *An deterministic (resp. probabilistic) algorithm is a deterministic (resp. probabilistic) Turing machine.*

Often the coin tosses in a probabilistic algorithm are considered as internal coin tosses. A second way to look at a probabilistic algorithm is to consider the output of the coin tosses as an additional input, which is supplied by an external coin-tossing device.

Given x, the output $A(x)$ of a probabilistic algorithm A is a random variable induced by the coin tosses. Let $A(x) = y$ denote the event "A outputs y on input x". By $\Pr[A(x) = y]$, we mean the probability of this event.

By $A(\cdot)$ we denote that the algorithm A has one input. By $A(\cdot, \ldots, \cdot)$ we denote that A has several inputs. $y \leftarrow A(x)$ denotes that y is obtained by running algorithm A on input x. In case A is deterministic, then this y is unique. If A is probabilistic (in which case we sometimes write $y \stackrel{R}{\leftarrow} A(x)$), then y is a random variable. If S is a set, then $y \leftarrow S$ (or sometimes $y \stackrel{R}{\leftarrow} S$) denotes that y was chosen form S uniformly at random.

Let b be a boolean function. The notation $(\{y_i \leftarrow A_i(x_i)\}_{i\in[1,n]} \| b(y_n))$ denotes the event that $b(y_n)$ is true after the sequential execution of A_i on input x_i, $i \in [1, n]$.

Definition 4 (Efficient Algorithm.) *An* efficient *algorithm or a* polynomial-time *algorithm is an algorithm whose worst-case running time function is of the form $O(n^k)$, where n is the input size and k is a constant.*

We use the shorthand notation "PPT" for *"probabilistic polynomial-time"* when describing an algorithm.

Next, we define what a two-party protocol is.

Definition 5 (Two-Party Protocol.) *A two-party protocol is a pair of interactive probabilistic Turing machines $(\mathcal{P}, \mathcal{V})$. An execution (or run) of the protocol $(\mathcal{P}, \mathcal{V})$ on input x (for $\mathcal{P}$) and y (for $\mathcal{V}$) is an alternating sequence of $\mathcal{P}$-rounds and $\mathcal{V}$-rounds, each producing a message to be delivered to the other party (except for the last $\mathcal{V}$-round). The sequence of such message is called the transcript of this run of the protocol.*

If, for all x and y, the length of such sequence, as well as the expected running time of $\mathcal{P}$ and $\mathcal{V}$, are polynomial in the length of x and y, then $(\mathcal{P}, \mathcal{V})$ is an *efficient* two-party protocol. By "$(\mathcal{P}(x) \leftrightarrow \mathcal{V}(y))$", we denote the probability space that assigns to a sequence of strings π the probability that a run of the $(\mathcal{P}, \mathcal{V})$ protocol, on input x and y, will produce π as transcript.

2.2.3 Relations and Languages

Computational problems are often modeled as decision problems: decide whether a given $x \in \{0,1\}^*$ belongs to a language $L \subseteq \{0,1\}^*$. First we recall the polynomial-time reduction among decision problems which is useful to compare their relative "hardness".

Definition 6 (Polynomial-time Reduction.) *Let L_1 and L_2 be two decision problems. L_1 is said to polytime reduce to L_2, written $L_1 \leq_P L_2$, if there is an algorithm that solves L_1 which uses, as a subroutine, an algorithm for solving L_2, and which runs in polynomial time if the algorithm for L_2 does.*

Let $R \subseteq \{0,1\}^* \times \{0,1\}^*$ be a binary relation. We say that R is *polynomially bounded* if there exists a polynomial Q such that $|w| \leq Q(|x|)$ holds for all (x, w) in R. Furthermore, R is an **NP**-*relation* if it is polynomially bounded and if there exists a polynomial-time algorithm for deciding membership pairs (x, w) in R. Let $L_R = \{x | \exists w \text{ such that } (x, w) \in R\}$ be the language defined by R. A language L is in **NP** if there exists an **NP**-relation $R_L \subseteq \{0,1\}^* \times \{0,1\}^*$ such that $x \in L$ if and only if there exists a w such that $(x, w) \in R_L$. Such a w is called a *witness* of the membership of x in L. The set of all witnesses of x is denoted by $R_L(x)$.

Definition 7 (The Classes P, NP and NPC.) *The complexity class* **P** *is the set of all decision problems that are solvable in polynomial time. The complexity class* **NP** *is the set of all decision problems for which a yes answer can*

be verified in polynomial time given some extra information, called a certificate. A decision problem L is said to be **NP**-*complete if* $L \in \mathbf{NP}$ *and* $L_1 \leq_P L$ *for every* $L_1 \in \mathbf{NP}$. *The class of all* **NP**-*complete problems is denoted by* **NPC**.

2.3 Algebra and Number Theory

Algebra and Number Theory are the mathematical foundation of Modern Cryptography. Numerous cryptographic algorithms are designed around results from them. They are also the cornerstone of (provable) security of cryptographic schemes.

We use the following notations. A prime number p is called a *safe prime* if $p = 2p' + 1$, such that p' is also a prime number. In this case, p' is known as a Sophie Germain prime. An integer n is called an *RSA modulus* if n is a product of two primes of equal size. An integer n is called a *safe-prime product*, if n is a product of two safe primes of equal size.

2.3.1 Groups

First recall the definition of a group (a cyclic group in particular) and some other related notions.

Definition 8 (Group.) *A group is a set G together with an associative binary operation $*$ on elements of G such that G contains an identity element for $*$ and every element has an inverse under $*$. If $*$ is commutative, the group is called abelian or commutative. Often, a group is denoted by $\langle G, * \rangle$ or simply by G. A group G is called finite if $|G|$ is finite. The number of elements of a finite group is called its order.*

Definition 9 (Cyclic Group.) *A group G is cyclic if there is $g \in G$ such that every element $a \in G$ can be written in the form of g^k for some $k \in \mathbb{Z}$.*

That is $G = \{g^i | i \geq 0\}$. *We call such* g *a generator of* G *and write* $\langle g \rangle = G$ *to indicate that* g *generates* G.

Definition 10 (Group Order.) *Let* G *be a group and* $a \in G$. *The order of* a, *denoted by* $\text{ord}(a)$, *is the smallest positive integer* n *such that* $a^n = 1$, *provided that such an integer exists. If such an* n *does not exist, then the order of* a *is defined to be* ∞.

Definition 11 (Subgroup.) *Let* $(G, *)$ *be a group. We say that* $(H, *)$ *is a subgroup of* G *if* $H \subseteq G$ *and* $(H, *)$ *is a group.*

The set of the integers modulo an integer n, denoted as $\mathbb{Z}_n$, together with addition modulo n constitutes an abelian group of order m. Another important group is $\mathbb{Z}_n^*$, formed by the positive integers smaller than n and relatively prime to n together with the multiplication modulo n. Finally denote by $QR(n)$ the cyclic group of quadratic residues modulo n with multiplication modulo n as the group operation.

The following definition is useful when we talk about group orders.

Definition 12 (Euler function.) *Let* n *be a positive integer. The Euler function* φ *is defined as the number of nonnegative integers* k *less than* n *which are relatively prime to* n:

$$\varphi(n) = |\{k | k \in [1, n-1] \text{ and } \gcd(k, n) = 1\}|.$$

The order of $\mathbb{Z}_n^*$ is given by the Euler totient function $\varphi(n)$. If n is prime, then $\varphi(n) = n - 1$. If $n = pq$ such that p and q are both prime, $\varphi(n) = (p-1)(q-1)$. In particular, when n is a prime $\varphi(n) = n - 1$. When $n = pq$ is a product of two primes, $\varphi(n) = (p-1)(q-1)$.

In this thesis, we always consider the group $QR(n)$ in which $n = (2p' + 1)(2q' + 1)$ is a safe-prime product. In such case, order of the group is $p'q'$.

2.3.2 Intractable Problems

Various cryptographic protocols rely their security on the intractability of one or more mathematical problems.

Definition 13 (RSA.) *Let n be a positive integer that is a product of two distinct odd primes p and q of the same size, e be a positive integer such that $gcd(e,(p-1)(q-1)) = 1$, and c be an integer. The RSA Problem is to find an integer m such that $m^e = c \pmod{n}$. The RSA Assumption says that there exists no PPT algorithm that can solve the RSA Problem in time polynomial in the size of n.*

Definition 14 (Discrete Logarithm (DL).) *Let G be a finite cyclic group generated by $g \in G$ of order $u = \#G$. The discrete logarithm of some element $a \in G$, denoted by $\log_g(a)$, is the unique integer x, $0 \leq x \leq u$, such that $a = g^x$. The* DL Problem *is to find $\log_g(a)$. The* DL Assumption *says that there exists no PPT algorithm that can solve the DL Problem, in time polynomial in the size of u.*

Definition 15 (Computational Diffie-Hellman (CDH).) *Let G be a cyclic group generated by $g \in G$ of order $u = \#G$. Given g, g^a and $g^b \in G$, the CDH Problem is to find the element $g^{ab} \in G$. The* CDH Assumption *says there exists no PPT algorithm that can solve the CDH Problem, in time polynomial in the size of u.*

Obviously, if the DL problem can be solved in polynomial time, then the DH problem can be solved in polynomial time. For some groups, the DH and the DL problems have been proved to be computationally equivalent. [18, 153, 154, 155].

Definition 16 (Decisional Diffie-Hellman (DDH).) *Let G be a cyclic group generated by g of order $u = \#G$. The* DDH Problem *is to distinguish between*

the distributions (g, g^a, g^b, g^c) *and* (g, g^a, g^b, g^{ab}), *with* $a, b, c \in_R \mathbb{Z}_u$. *The* DDH Assumption *says there exists no PPT algorithm solve the DDH Problem, in time polynomial in the size of* u.

The DDH problem was first mentioned in [7], although there are earlier cryptographic systems that implicitly rely on the hardness of this problem, e.g. [34].

Definition 17 (Strong RSA.) *Let* $n = pq$ *be an RSA modulus. Let* G *be a cyclic subgroup of* $\mathbb{Z}_n^*$ *of order* $u = \#G$, $\lceil \log_2(\#G) \rceil = \ell_G$. *Given* n *and* $z \in G$, *the* Strong RSA Problem *is to find* $u \in G$ *and* $e \in \mathbb{Z}_{>1}$ *such that* $z = u^e \bmod n$. *The* Strong RSA Assumption *says that there exists no PPT algorithm that can solve the Strong RSA Problem, in time polynomial in the size of* u.

Remark: If n is a safe RSA modulus, it is a good habit to restrict operation to the subgroup of quadratic residues modulo n, i.e. the cyclic subgroup $QR(n)$ generated by an element of order $p'q'$. This is because the order $p'q'$ of $QR(n)$ has no small factors.

The Strong RSA Assumption was independently introduced by Barić and Pfitzmann [4] and by Fujisaki and Okamoto [51].

In fact, all of the intractable problems above are only believed to be intractable without proof or disproof. However, one should feel fairly confidence on their hardness because no attacks has been successful despite the fact that they have been studied (or attack) for so many years. In the following, we briefly describe, as an example, the attacks on the DL Problem.

Known attacks on the DL Problem can be classified into *generic* or *specific*. Generic attacks do not use the properties of the underlying group and can thus be applied to any group while specific methods exploits group-specific structure. Examples of generic methods are Shanks' "Baby-Step Giant-Step" method and Pollard's "Lambda" and "Rho" methods. The amount of work

that needs to be done to solve the discrete logarithm with a generic method grows exponentially in the size of the input. This makes groups on which no attacks other than generic ones are known suitable for the design of DL-based cryptographic protocols.

Examples of specific methods include Index Calculus and Number Field Sieve algorithms. Simply speaking, these algorithms require a factor base which is a small suitable set of elements, and a way to decompose random group elements into elements of the factor base. As a result, these methods can only be applied to groups in which such a suitable factor base and decomposition method exist. The only groups that are known so far that satisfy the stringent requirements are multiplicative groups of finite fields and class groups of imaginary quadratic fields.

2.4 Cryptographic Primitives

2.4.1 Public-Key Encryption

Encryption schemes aim at allowing one party to send data to another in a confidential way. To do this, a sender "encrypt" a message into a ciphertext, which is then sent to the receiver, such that only the intended receiver is capable of retrieving the original message by "decrypting" the ciphertext.

A public key encryption scheme is a triple of polynomial-time algorithms (G, E, D). The probabilistic *Key Generation* algorithm G generates a secret key x and a corresponding public key y for an entity when input the system's parameters. The probabilistic *Encryption* algorithm E takes a message m and a public key y as input and outputs a ciphertext c. The deterministic *Decryption* algorithm D, on input of a ciphertext c and a secret key x, outputs a message m. A public key encryption scheme must be correct, i.e. for all messages m and all key pairs (x, y) output by G, it holds that $D(E(m, y), x) =$

m.

The security goal of an encryption is defined as the indistinguishability of ciphertexts under adaptive chosen ciphertext attack (IND-CCA2), which roughly means the following. An adversary is given decryption oracle to which he can access adaptively. The adversary is then expected to give two plaintexts of his choice and a ciphertext of either of the plaintexts is returned with equal probability. The adversary is asked to distinguish the plaintext behind the ciphertext. Of course, the adversary is not allowed to query the decryption oracle on the challenge ciphertext. If the adversary cannot decide correctly (non-negligibly) better than pure guessing, then the ciphertext reveals no (non-negligible) information about the plaintext behind.

Another popular security goal of an encryption scheme is the so-called NM-CCA2, which stands for non-malleability under adaptive chosen ciphertext attack. Non-malleability means that an adversary given a challenge ciphertext is unable to obtain a different ciphertext such that the plaintexts underlying these two cihpertexts are "meaningfully related".

There exists some other weaker security models in which the attacker is given less power. For example, under chosen plaintext attack (CPA) the adversary can obtain ciphertext of any plaintext; under non-adaptive chosen ciphertext attack (CCA1) the adversary can get access to an oracle for the decryption function only for the period of time preceding his being given the challenge ciphertext. In other words, adversary's queries to the decryption oracle cannot depend on the challenge ciphertext. These models are too weak to model the real world and should not be used as schemes proven secure under these models are still not guaranteed to be secure in practice.

We give two examples of encryption schemes below.

RSA Encryption

The scheme is due to [90] and works as follows.

- Key Generation: Let p and q be two large primes such that $p-1$ and $q-1$ are not smooth and $n = pq$. Let e be an integer satisfying $gcd(e, \varphi(n)) = 1$. The public key of a recipient Bob is the pair (n, e) and his secret key is the triple (p, q, d), where d satisfies $de = 1 \pmod{\varphi(n)}$.

- Encrypt: To encrypt a message $m \in [0, n-1]$ for Bob, a sender Alice computes and sends to Bob the ciphertext $c := m^e \bmod n$.

- Decrypt: Bob can recover m using the secret value d by computing: $m := c^d \bmod n$.

The security of the scheme is based on the RSA Assumption. Although the encryption scheme above is simple to understand and widely deployed in the old days, it is not secure against the IND-CCA2 model, which means the scheme should not be used for security reasons. The OAEP techniques introduced a couple of years ago wraps up the RSA encryption scheme as the core and is the first IND-CCA2 secure RSA-based encryption scheme.

ElGamal Encryption

This is due to ElGamal [ElG85a,ElG85b]. Let G be $\mathbb{Z}_p^*$ where p is a large prime. Let $g \in G$ be a generator of G.

- Key Generation: Randomly pick the secret key $x \in_R \mathbb{Z}_p$ and compute the public key $y := g^x$.

- Encryption: To encrypt a message $m \in G$, choose an r randomly in $\mathbb{Z}_p$ and computes the ciphertext $(A, B) = (g^r, y^r m)$.

- Decryption: Recover m by computing $\frac{B}{A^x}$.

The security is based on the assumed intractability of the DHP. Again, the scheme above is not secure in the sense of IND-CCA2. Cramer and Shoup provided an IND-CCA2 secure ElGamal-based encryption scheme.

2.4.2 Identification Protocols

An identification protocol aims at allowing a prover Peggy to convince a verifier Victor of her identity. The goal of such schemes is that nobody except than Peggy, not even Victor, is able to represent Peggy even he/she listened to previous identification of Peggy to Victor before. A general way to do identification is as follows: Peggy has a secret key only known to her and a public key known to Victor. Victor concludes that a person is Peggy if he/she can prove to Victor that he/she knows Peggy's secret key. Of course, the process of such a proof has to be "zero-knowledge" or otherwise information about the secret key may leak, possibly enabling others to impersonate Peggy after listening to the conversation between Peggy and Victor.

The Schnorr Identification Protocol

The Schnorr protocol allows a prover to prove that he knows the discrete logarithm of a group element. In particular if the public key y and the secret key x are such that $y = g^x$, then the Schnorr protocol allows one to prove that he knows the secret key that corresponds to some public key.

The protocol is done as follows. Both the prover and the verifier are given the common input the description of a finite cyclic group G of order q, an element $g \in G$ that generates G, an element $y \in G$ (the public key), the additional input to the prover is an element $x \in [1, q]$ such that $y = g^x$ (the secret key). The prover picks $r \in_R \mathbb{Z}_q$, computes $t := g^r$ and sends it to the verifer. The verifier picks $c \in_r \{0,1\}^k$ and sends it to the prover. The prover computes $s := r - cx$ and sends s to the verifier. The verifier outputs yes if $g = g^s y^c$, and no otherwise.

It is proved that if a verifier outputs yes, then the probability that the prover actually does not know the secret is 2^{-k}. It is also proved that if the verifier honestly generates the challenge, then the protocol reveals nothing

about he secret to the verifier.

2.4.3 Digital Signatures

A digital signature scheme is a triple of polynomial-time algorithms (G, S, V). The probabilistic *Key Generation* algorithm G generates a secret key x and a corresponding public key y for a signer on input of the system's parameters. The probabilistic *Signing* algorithm S takes a message m and a secret key x as input and outputs a signature σ of m. The deterministic *Verification* algorithm, on input of a message m, a signature σ, and the public key y, outputs either true or false. A digital signature scheme must be correct, i.e. for all messages m, for all key pairs (x, y) output by G, it holds that $V(m, S(m, x), y) = 1$.

A secure digital signature scheme must be unforgeable. The previous statement is vague because unforgeability can be defined in a number of ways depending both on the attacks an adversary mounts and on the forgery attempted.

In regards to attacks, they range from (1) a *known plaintext attack* (in which the adversary is given a set of signatures and the respective messages), to (2) a *chosen plaintext attack* (where the adversary chooses a list of messages and asks the signer for their signatures), to (3) an *adaptive chosen plaintext attack* (in which the adversary uses the signer as an "oracle", asking for signatures on message of his choice).

In terms of forgery, there are several levels of success for an attacker: (1) *existential forgery* means the adversary succeeds in obtaining a signature on one message, which may not be of his choice, or even meaningful; (2) *selective forgery* means the adversary obtains a signature on a message of his choice; (3) *universal forgery* means the adversary, although unable to find the secret key of the signer, is able to forge the signature of any message; and (4) *total*

break means the adversary succeeds in obtaining the signer's private key.

A signature scheme is secure if it is existentially unforgeable under adaptive chosen message attack, meaning an adversary can not succeed in the least significant way, even he is mounting the strongest attack.

The following are examples of digital signature schemes.

RSA Signature

The RSA signature scheme is directly derived from the RSA encryption scheme by reversing the roles of encryption and decryption [90]. Let H be a collision-resistant hash function that maps $\{0,1\}^*$ to $\mathbb{Z}_n$.

- Key Generation: Randomly pick a prime p and a generator of $\mathbb{Z}_p^*$. Randomly pick an $x \in [0, p-2]$ and compute $y := g^x \bmod p$. The public key is (g, p, y), the secret key is (g, p, x).

- Signing: Randomly pick $r \in [1, p-2]$ that is prime to $p-1$. Compute $s_1 := g^r \bmod p$ and $s_2 := (H(m) - xs_1)r^{-1} \bmod (p1-)$. The signature is (s_1, s_2).

- Verification: A signature verifies if $y^{s_1} s_1^{s_2} \equiv g^{H(m)} \pmod{p}$.

ElGamal Signature

Let H be a collision-resistant hash function that maps $\{0,1\}^*$ to $\mathbb{Z}_n$.

- Key Generation: Randomly pick two primes p and q of equal size such that $p-1$ and $q-1$ are not smooth, and choose an integer e such that $\gcd(e, \varphi(n)) = 1$, where $n = pq$. A signer's public key is the pair (n, e) and his secret key is the triple (p, q, d), where $d = e^{-1} \pmod{\varphi(n)}$.

- Signing: An RSA signature of a message $m \in \{0,1\}^*$ for the public key (n, e) is computed as $\sigma := H(m)^d \pmod{n}$.

- Verification: A signature verifies if $s^e \equiv H(m) \pmod{n}$.

A variant of the ElGamal signature scheme, called Digital Signature Algorithm (DSA), is proposed as a standard by the U.S. National Institute of Standards and Technology (NIST). The Digital Signature Standard (DSS) is the first digital signature scheme recognized by any government.

Schnorr Signature

The Schnorr signature is a variant of the ElGamal signature scheme. The Schnorr signature scheme is an example of the construction of a signature scheme from an identication protocol [203]. Compared with the ElGamal signature scheme, the Schnorr scheme provides shorter signatures for the same level of security.

- Key Generation: Randomly select the secret key $x \in \mathbb{Z}_q$. Compute the public key $y = g^x$.
- Signing: choose $r \in_R \mathbb{Z}_q$, compute $c := H(m||g^r)$. Compute $s := r - cx \pmod{q}$. The signature is (c, s).
- Verification: A signature verifies if $c = H(m||g^s y^c)$.

2.4.4 Hash Functions

A hash function is an efficiently computable function mapping binary strings of arbitrary finite length to binary strings of a fixed length ℓ:

$$H : \{0,1\}^* \rightarrow \{0,1\}^\ell.$$

As long as cryptographic use is concerned, a hash function may have the following potential security properties:

- *(One-wayness.)* For a given c, it is hard to find an x such that $H(x) = c$.

- *(Weak collision resistance.)* For a given x, it is hard to find an $x' \neq x$ such that $H(x) = H(x')$.

- *(Strong collision resistance.)* It is hard to find a pair (x, x') with $x \neq x'$ such that $H(x) = H'(x)$ if H is chosen at random from a family of hash-functions.

In the above, strong collision resistance implies weak collision resistance which in turn implies one-wayness. It is sufficient to assume all the hash functions appeared in this thesis to be weak collision resistant.

Today, an output size of 160 bits (or larger) seems to have a reasonable security.

Examples of hash functions used in Cryptography: MD4 [88], MD5 [89], SHA-1 [83], and etc.

The Random Oracle Model

The Random Oracle Model (ROM) is a paradigm that acts as a bridge between cryptographic theory and cryptographic practice. The idea of ROM firstly appeared in the paper written by Fiat and Shamir [46] and was formulated by Bellare and Rogaway [7]. Canetti *et.al* [26, 27] presented the gap between the model and the real implementations. [7] raised an implicit philosophy behind the use of a random oracle to an explicitly articulated paradigm with which maintains the benefits to practice.

In practice, it is no formal definition to the hash functions (pseudorandom function family) but we capture a number of nice properties which it seems to possess. In the ROM, we assume hash functions are random functions and are publicly accessible by all parties. Random oracle, H, is an object to instantiate all hash functions in the model and reply all queries from the parties. A polynomial time algorithm cannot distinguish the query replied from

a real world or the random oracle simulated by a function. It has following properties:

1. H is assumed to be a random function such that given an input, any party cannot guess the output with non-neglibible probability.
2. H is a one-way function. Given an output, it is difficult to figure out the preimage of it.
3. $H : \{0,1\}^* \rightarrow \{0,1\}^\infty$
4. H is collision resistant so that given x, y, where $x \neq y$, $H(x) = H(y)$ with negligible probability.
5. Given same inputs x and y, where $x = y$, $H(x) = H(y)$.
6. All parties in the model must query H for getting random values. They cannot distinguish the values generated by H from the real hash function.

It is obvious that there exists no hash function which behaves a random function. ROM has an assumption that the oracle is susceptible to attack but in reality, it may not be true. Therefore, a protocol which is said to be provably secure in the random oracle model may be insecure in practice. Despite of its impractical assumptions, the paradigm is useful to yield an efficient solution to prove the security of a protocol. It is better than no proof shown.

2.4.5 Zero-Knowledge Proof of Knowledge

Zero-knowledge Proof of Knowledge is a crucial primitive in Cryptography and its application can be widely found in cryptographic protocols. Speaking at a high level, it is a protocol run between two parties, allowing one to prove to another the knowledge of some secret, without leaking any information about the secret. We need the following definitions to make our description precise.

Definition 18 (Ensembles.) *Let I be a countable index set. An ensemble indexed by I is a sequence $X = \{X_i\}_{i \in I}$, where each X_i is a random variable over $\{0,1\}^*$.*

Definition 19 (Indistinguishability.) *Let $L \in \{0,1\}^*$ be a language and let $A = \{A(x)\}_{x \in L}$ and $B = \{B(x)\}_{x \in L}$ be two ensembles of random variables indexed by strings $x \in L$. We say that the ensembles A and B are*

- perfectly indistinguishable *if for all $x \in L$ the random variables $A(x)$ and $B(x)$ are identically distributed.*
- statistically indistinguishable *if their statistical difference is negligible, or more technically, if for every polynomial $p(\cdot)$ and for all sufficiently long $x \in L$ it holds that*

$$\sum_{\alpha \in \{0,1\}^*} |\Pr(A(x) = \alpha) - \Pr(B(x) = \alpha)| < \frac{1}{p(|x|)}.$$

- computationally indistinguishable *if no efficient algorithm exists that can distinguish them, i.e. for every PPT algorithm D, for every polynomial $p(\cdot)$ and for all sufficiently long $x \in L$,*

$$|\Pr(D(x, A(x)) = 1) - \Pr(D(x, A(x)) = 0)| < \frac{1}{p(|x|)}.$$

Definition 20 (Zero-knowledgeness.) *An interactive protocol $(\mathcal{P}, \mathcal{V})$ is perfect/statistical/computational zero-knowledge, if for every PPT verifier $\tilde{\mathcal{V}}$ there exists a probabilistic expected polynomial time simulator $\mathcal{S}_{\tilde{\mathcal{V}}}$ so that the two ensembles*

$$\{[\tilde{\mathcal{V}}, \mathcal{P}](x)\}_{x \in L} \text{ and } \{\mathcal{S}_{\tilde{\mathcal{V}}}(x)\}_{x \in L}$$

are perfectly/statistically/computationally indistinguishable.

Definition 21 (Honest-Verifier Zero-knowledgenes.) *An interactive protocol $(\mathcal{P}, \mathcal{V})$ is perfect (statistical/computational) honest-verifier zero-knowledge, if there exists a probabilistic expected polynomial-time simulator $\mathcal{S}_{\mathcal{V}}$ so that the two ensembles*

$$\{[\mathcal{V}, \mathcal{P}](x)\}_{x \in L} \text{ and } \{\mathcal{S}_{\mathcal{V}}(x)\}_{x \in L}$$

are perfectly (statistically/computationally) indistinguishable.

Definition 22 (Proof of Knowledge.) *Let $R \subseteq \{0,1\}^* \times \{0,1\}^*$ be a polynomially bounded binary relation and let L_R be the language defined by R. The protocol $(\mathcal{P}, \mathcal{V})$ is said to be an interactive proof of knowledge for the relation R if the following are satisfied:*

- (Completeness.) *On common input x, if the honest prover $\mathcal{P}$ gets as private input w such that $(x, w) \in R$, then the verifier $\mathcal{V}$ always accepts.*
- (Validity.) *Let $\mathcal{K}$ be an PPT algorithm known as the knowledge extractor that gets as inputs an $x \in L_R$ and rewindable black-box access to a prover and attempts to compute w such that $(x, w) \in R$. Soundness means the existence of $\mathcal{K}$ such that the following hold. For any prover $\tilde{\mathcal{P}}$, for all polynomials $p(\cdot)$ and for all sufficiently large $x \in L_R$, $\mathcal{M}$ will output a a w such that*

$$\Pr((x, w) \in R) \geq \Pr(\tilde{\mathcal{P}} \text{ convinces } \mathcal{V} \text{ on } x) - \frac{1}{p(|x|)}.$$

Σ-Protocols

A Σ-protocol for an **NP**-relation R is an efficient 3-round two-party protocol, such that for every input (x, y) to $\mathcal{P}$ and y to $\mathcal{V}$, the first $\mathcal{P}$-round yields a commitment message t, the subsequent $\mathcal{V}$-round replies with a random challenge message c, and the last $\mathcal{P}$-round concludes by sending a response message s. At the end of a run, $\mathcal{V}$ outputs a 0/1 value, functionally dependent on y and

the transcript $\pi \doteq (t, c, s)$ only; a transcript is valid if the output of the honest verifier is 1.

Additionally, we require that a Σ-protocol satisfies:

- *(Special Soundness.)* There is an efficient algorithm $\mathcal{K}$ (called a Knowledge Extractor) that on input any $y \in L_R$ and any pair of valid transcripts with the same commitment message (t, c, s) and (t, c', s') outputs x such that $(x, y) \in R$.

- *(Special Honest-Verifier Zero-Knowledge (Special HVZK).)* There is an efficient algorithm $\mathcal{S}$ (called a Simulator) that on input $y \in L_R$ and any challenge message c, outputs a pair of commitment/response messages t, s, such that the transcript $\pi \doteq (t, c, s)$ is valid, and it is distributed according to the probability distribution $(\mathcal{P}(x, y) \leftrightarrow \mathcal{V}(y))$, for any y such that $(x, y) \in R$.

It can be shown that a Σ-protocol $(\mathcal{P}, \mathcal{V})$ with special soundness is a proof of knowledge in the sense of Bellare and Goldreich [5] with knowledge error $2^{-|c|}$, by the results of Damgård and Pfitzmann [38].

One way to interpret the knowledge error is to think of it as the probability that one can convince the verifier without knowing a correct witness. To have a probability higher than that one must have some ability to actually compute the witness.

Due to [37], if a Σ-protocol is HVZK, the protocol is perfectly witness indistinguisable (WI) [45]. Although HVZK by itself is defined with respect to a very much restricted verifier, i.e. an honest one, this means that if for a given instance α there are at least two witnesses β, then even an arbitrarily powerful and malicious verifier cannot distinguish which witness the prover uses.

Signature of Knowledge

Recall that in the Random Oracle Model, protocol participants have access to a random oracle, which is an entity that initially chooses (in private) a random function $R : \{0,1\}^l \to \{0,1\}^t$ for some $l, t \in \mathbb{N}$. Then any player can send any l-bit string a to the oracle to obtain $R(a)$. Since R was completely random, $R(a)$ is a uniformly chosen t-bit string independent of a. Also, knowing $R(a)$ gives no advantage in predicting the value $R(b)$ for any $b \neq a$. However, every time someone sends a to the oracle, the answer will be the same value $R(a)$.

With such an oracle, it is possible for the prover in a Σ-protocol to run the protocol without communicating with the verifier. Instead, the prover replaces the verifier's random choice of challenge by sending the first message to the oracle, and using the response as challenge. If this generated the "conversation" (t, c, s), the prover can send (t, s) to the verifier in one message. The verifier calls the oracle with t as input to get the value of c, and checks the answer s as it would have done normally.

The prover cannot get an oracle response on t without calling the oracle, and hence he has no information about c before he has sent t. In this regard, the situation is equivalent to talking to a real verifier. Of course, a adversarial prover is now free to call the oracle as many times as he wants, hoping that he can obtain a challenge that he can answer. But if the number of challenges is exponentially large, this is infeasible for a polynomial-time adversarial prover.

Using the random oracle has also the effect of preventing a verifier from cheating, or forcing the verifier to be honest. This is because the challenges are always randomly and independently chosen, just like the honest verifier would choose them. Therefore, the simulator is allowed to decide what the oracle outputs should be, as long as they have the same distribution indistinguishable from that in real life. As a result, one can simply pick c at random and run the normal simulator $\mathcal{S}$ to get a protocol conversation (t, c, s), and assign the

oracle's response on input a to be e, and output (t, s). This type of construction has been used to make secure signature schemes from Σ-protocols: to generate keys we sample a pair (y, x) in certain hard relation R, and let the image y be the public key, and its witness x be the private key. To sign a message M, the signer executes the Σ-protocol in the role of the prover and computes the first message t. He then calls the random oracle with (t, M) as input, and takes the answer c as the challenge. Using his knowledge of x, he can compute the answer s. The signature is then the pair (t, s). In the random oracle model, one can then prove that breaking this signature scheme is as hard as computing x from y.

Using the techniques introduced in [46, 47], every three-round Proof of Knowledge Protocols (PoKs) that is Honest-Verifier Zero-Knowledge (HVZK) can be turned into a signature scheme by setting the challenge to the hash value of the commitment together with the message to be signed. Such schemes are simulatable and proven secure by [86] against existential forgery under adaptively chosen message attack [55] in the random oracle model [7]. Simulatability means that the distribution of the strings that can be efficiently generated without knowledge of the secret signing key are indistinguishable from the distribution of the actual signatures.

A signature of knowledge allows a signer to prove the knowledge of a secret with respect to some public information noninteractively. The signer can also tie his knowledge of a secret to a message being signed. Following [25], we call these signature schemes "signatures based on proofs of knowledge", SPK for short. Note that there always exists a corresponding HVZK PoK protocol for every SPK.

Sometimes we describe proofs of knowledge not as protocols but rather as signature schemes derived from these protocols, since we most often use them as such. However, the reader should keep in mind that there always exists a corresponding protocol being a proof of knowledge.

2.4.6 Accumulators

An *accumulator family* is a pair $(\{F_\lambda\}_{\lambda\in\mathbb{N}}, \{X_\lambda\}_{\lambda\in\mathbb{N}})$, where $\{F_\lambda\}_{\lambda\in\mathbb{N}}$ is a sequence of families of functions such that each $f \in F_\lambda$ is defined as $f : U_f \times X_f^{\text{ext}} \to U_f$ for some $X_f^{\text{ext}} \supseteq X_\lambda$ and additionally the following properties are satisfied:

- *(Efficient generation.)* There exists an efficient algorithm G that on input a security parameter 1^λ outputs a random element f of F_λ, possibly together with some auxiliary information a_f.
- *(Efficient evaluation.)* Any $f \in F_\lambda$ is computable in time polynomial in λ.
- *(Quasi-commutativity.)* For all $\lambda \in \mathbb{N}$, $f \in F_\lambda$, $u \in U_f$, $x_1, x_2 \in X_\lambda$, $f(f(u, x_1), x^2) = f(f(u, x_2), x_1)$.

We will refer to $\{X_\lambda\}_{\lambda\in\mathbb{N}}$ as the *value domain* of the accumulator. For any $\lambda \in \mathbb{N}$, $f \in F_\lambda$ and $X = \{x_1, \ldots, x_s\} \subset X_\lambda$, we will refer to $f(\ldots f(u, x_1) \ldots, x_s)$ as the *accumulated value* of the set X over u. Due to quasi-commutativity, such value is independent of the order of the x_i's and will be denoted by $f(u, X)$.

Definition 23 An accumulator is said to be collision resistant if for any $\lambda \in \mathbb{N}$ and any PPT algorithm $\mathcal{A}$,

$$\Pr\left[\begin{array}{c} f \xleftarrow{R} F_\lambda; \\ u \xleftarrow{R} U_f; \\ (x, w, X) \xleftarrow{R} \mathcal{A}(f, U_f, u) \end{array} \middle| \begin{array}{c} (X \subseteq X_\lambda) \wedge (w \in U_f) \wedge \\ (x \in X_f^{\text{ext}} \backslash X) \wedge (f(w, x) = f(u, X)) \end{array}\right] = \nu(\lambda),$$

where $\nu(\lambda)$ is some negligible function in λ.

For $\lambda \in \mathbb{N}$ and $f \in F_\lambda$, we say that $w \in U_f$ is a *witness* for the fact that $x \in X_\lambda$ has been accumulated with $v \in U_f$ (or simply that w is a witness for

x in v) whenever $f(w, x) = v$. We extend the notion of witness for a set of values $X = \{x_1, \ldots, x_s\}$ in a straightforward manner.

An efficient construction of a collision-resistant accumulator was presented in [20], based on earlier work by [4] and [10].

Accumulators with One-Way Domain

An *accumulator with one-way domain* is a quadruple $(\{F_\lambda\}_{\lambda\in\mathbb{N}}, \{X_\lambda\}_{\lambda\in\mathbb{N}}, \{R_\lambda\}_{\lambda\in\mathbb{N}})$, such that the pair $(\{F_\lambda\}_{\lambda\in\mathbb{N}}, \{X_\lambda\}_{\lambda\in\mathbb{N}})$ is a collision-resistant accumulator, and each R_λ is a relation over $X_\lambda \times Z_\lambda$ with the following properties:

- *(Efficient verification.)* There exists an efficient algorithm D that on input $(x, z) \in X_\lambda \times Z_\lambda$, returns 1 if and only if $(x, z) \in R_\lambda$.
- *(Efficient sampling.)* There exists a probabilistic algorithm W that on input 1^λ returns a pair $(x, z) \in X_\lambda \times Z_\lambda$ such that $(x, z) \in R_\lambda$. We refer to z as a *pre-image* of x.
- *(One-wayness.)* It is computationally hard to compute any pre-image z' of an x that was sampled with W. Formally, for any PPT algorihtm $\mathcal{A}$,

$$\Pr[(x, z) \xleftarrow{R} W(1^\lambda); z' \xleftarrow{R} \mathcal{A}(1^\lambda, x) | (x, z') \in R_\lambda] = \nu(\lambda).$$

Dodis gave an efficient construction of collision resistant accumulators with one-way domain in [43], based on [20].

For $\lambda \in \mathbb{N}$, the family F_λ consists of the exponentiation functions modulo λ-bit rigid integers

$$f : QR(n) \times \mathbb{Z}_{n/4} \to QR(n)$$

$$f : (u, x) \mapsto u^x \mod n$$

where n is a λ-bit rigid integer.

The accumulator domain $\{X_\lambda\}_{\lambda\in\mathbb{N}}$ is defined by:

$$X_\lambda \doteq \{e \text{ prime}|(\frac{e-1}{2} \in \mathsf{RSA}_\ell) \wedge (e \in S(2^\ell, 2^\mu))\}$$

where $S(2^\ell, 2^\mu)$ is embedded within $(0, 2^\lambda)$ with $\lambda - 2 > \ell$ and $\ell/2 > \mu + 1$. The pre-image domain $\{Z_\lambda\}_{\lambda\in\mathbb{N}}$ and the one-way relation $\{R_\lambda\}_{\lambda\in\mathbb{N}}$ are defined as follows:

$$Z_\lambda \doteq \{(e_1, e_2)|e_1, e_2 \text{ are distinct } \ell/2\text{-bit primes and } e_2 \in S(2^{\ell/2}, 2^\mu)$$

$$R_\lambda \doteq \{(x, (e_1, e_2)) \in X_\lambda \times Z_\lambda|(x = 2e_1e_2 + 1)\}$$

2.4.7 Public Key Infrastructure

An X.509 compliant Public Key Infrastructure (PKI) is composed of three main entities:

- *Certification Authority (CA).* the core of a PKI, it is a trusted system that warrants the binding between a public key and its owner by means a certificate, which it signs with its private key and makes accessible to all users. Certificate management is completed with certificate revocation in case of accidental events, such as key compromise or less, that force the revocation of the certificate before its natural expiration date. A CA performs the following basic operations: (1) issuing end user certificates; (2) issuing cross-certificates for other CAs; (3) processing certificate revocation requests from end users and RAs; (4) generating periodic Certificate Revocation List (CRL), or updates thereof.

- *Registration Authority (RA).* An optional system component to which the CA may delegate certain functions, such as verifying users' identity or performing the proof of possession of the private key, with the purpose

of reducing the accesses to the CA. A certificate signed by the CA guarantees its authenticity. All the communications with the CA are digitally signed. An RA performs the following operations: (1) vouching for the identity of entities requesting certification of their keys; (2) identity verification by requiring the entity to appear at the RA personally with a physical token or through out-of-band mechanisms; (3) verification of the user's possession of the private key; (4) signing an electronic certificate request and sending it to the appropriate CA; (5) requesting certificate revocations for user certificates issued by CAs that have accredited it.

- *Repository or Directory Server.* A system, or a collection of distributed systems, that stores certficates and CRLs as distribution center for users. It does not need to be trusted because the CA signs the objects it deals with. It usually satisfies three types of requests: (1) add requests, performed by the CA to publish certificates and CRL's; (2) modify requests, performed by the CA to change object attributes; (3) download requests, performed by any entity wishing to verify the validity of a certificate.

In a large scale PKI there might be various CAs, RAs and Repositories. Each CA has one or more RAs that refer to it and can publish data in one or more repositories. CAs can be hierarchically organized or networked. In hierarchical models a CA delegates trust to subordinate CAs when it certifies them. Trust delegation starts at a root CA, which is trusted at every node of the infrastructure. In networked models, also known as cross certified models, trust is established between two CAs in a peer to peer relationships. The daily activity of issuing and revoking certificates is managed by a CA in the same way in both models.

2.5 Zero Knowledge Proof of Knowledge Protocols in Groups of Unknown Order

2.5.1 The Algebraic Setting

Let N be a λ-bit safe prime product and let G be the finite cyclic group $QR(N)$. Let k an integer, and let g, h, $g_1, \ldots, g_k \in G$ be generators of G such that computing discrete logarithms of any group element (apart from the the identity element) with respect to one of the generators is infeasible. The generators are chosen in random, so that discrete logarithms of no generator with respect to another are known.

Fujisaki and Okamota showed in [51] that, under the Strong RSA Assumption, the standard proofs of knowledge protocols that work for a group of known order are also proofs of knowledge in this setting. The first example is given as follows.

Definition 24 (Discrete Logarithm in $QR(N)$.) *Let $y, g \in G$. A pair $(c, s) \in \{0,1\}^k \times \pm\{0,1\}^{\epsilon(\ell_G+k)+1}$ verifying $c = \mathcal{H}(y||g||g^s y^c||m)$ is a signature of knowledge of the discrete logarithm of $y = g^x$ w.r.t. base g, on a message $m \in \{0,1\}^*$.*

The party in possession of the secret $x = \log_g y$ is able to compute the signature by choosing a random $t \in \pm\{0,1\}^{\epsilon(\ell_G+k)}$ and then computing c and s as:

$$c = \mathcal{H}(y||g||g^t||m) \text{ and } s = t - cx \text{ in } \mathbb{Z}.$$

The security of the above has been proven in the random oracle model [7] under the strong RSA assumption in [18, 51, 52]. That is, if $\epsilon > 1$, then the corresponding interactive protocols are statistical (honest-verifier) zero-knowledge proofs of knowledge.

In the following, we are going to describe some three-move interactive

HVZK PoK protocols that we will use as basic building blocks for our forward secure threshold ring signature scheme. These protocols all work in finite cyclic groups of quadratic residues modulo safe prime products. For each $i = 1, \ldots, n$, let N_i be a safe-prime product and define the group $G_i \doteq QR(N_i)$ such that its order is of length $\ell_i - 2$ for some $\ell_i \in \mathbb{N}$. Also let g_i, h_i be generators of G_i such that their relative discrete logarithms are not known.

Let $1 < \epsilon \in \mathbb{R}$ be a parameter and let $H : \{0,1\}^* \rightarrow \mathbb{Z}_q$ be a strong collision-resistant hash function, where q is a κ-bit prime for some security parameter $\kappa \in \mathbb{N}$. Define $\mathcal{N} \doteq \{1, \ldots, n\}$ and $\Gamma_i \doteq \{-2^{\ell_i}q, \ldots, (2^{\ell_i}q)^{\epsilon}\}$.

2.5.2 Proving the Knowledge of Several Discrete Logarithms

This protocol is a straightforward generalization of the protocol for proving the knowledge of a discrete logarithm over groups of unknown order in [18]. This allows a prover to prove to a verifier the knowledge of n discrete logarithms $x_1, \ldots, x_n \in \mathbb{Z}$ of elements $y_1, \ldots, y_n$ respectively and to the bases $g_1, \ldots, g_n$ respectively. Using the notation in [25], the protocol is denoted by:

$$PK\{(\alpha_1, \ldots, \alpha_n) : \bigwedge_{i=1}^{n} y_i = g_i^{\alpha_i}\}.$$

A prover $\mathcal{P}$ knowing $x_1, \ldots, x_n \in \mathbb{Z}$ such that $y_i = g_i^{x_i}$ for all $i = 1, \ldots, n$ can prove to a verifier $\mathcal{V}$ his/her knowledge as follows.

- (Commit.) $\mathcal{P}$ chooses $r_i \in_R \mathbb{Z}_{(2^{\ell_i}q)^{\epsilon}}$ and computes $t_i \leftarrow g_i^{r_i}$ for all $i = 1, \ldots, n$. $\mathcal{P}$ sends $(t_1, \ldots, t_n)$ to $\mathcal{V}$.
- (Challenge.) $\mathcal{V}$ chooses $c \in_R \mathbb{Z}_q$ and sends it to $\mathcal{P}$.
- (Response.) $\mathcal{P}$ computes, for all $i = 1, \ldots, n$, $s_i \leftarrow r_i - cx_i$ (in $\mathbb{Z}$). $\mathcal{P}$ sends $(s_1, \ldots, s_n)$ to $\mathcal{V}$.

$\mathcal{P}$ verifies by checking, for all $i = 1, \ldots, n$, if $t_i \stackrel{?}{=} g_i^{s_i} y_i^c$.

Theorem 1 If the Strong RSA assumption holds, the protocol stated in Sec. 2.5.2 is an HVZK PoK protocol.

Proof 1 We omit the proof as it is a straightforward extension of the proof of Lemma 1 in [18]. □

As noted before, the protocol can be turned into a signature scheme by replacing the challenge by the hash of the commitment together with the message M to be signed: $c \leftarrow H((g_1, y_1)|| \ldots ||(g_n, y_n)||t_1|| \ldots ||t_n||M)$. In this case, the signature is $(c, s_1, \ldots, s_n)$ and the verification becomes:

$$c \stackrel{?}{=} H((g_1, y_1)|| \ldots ||(g_n, y_n)||g_1^{s_1} y_1^c|| \ldots ||g_n^{s_n} y_n^c||M).$$

Following [25], we denote this signature scheme by:

$$SPK\{(\alpha_1, \ldots, \alpha_n) : \bigwedge_{i=1}^{n} y_i = g_i^{\alpha_i}\}(M).$$

2.5.3 Proving the Knowledge of a Representation

This protocol is a generalization (using the method described in [32]) of the protocol for proving the knowledge of a representation in [18]. This allows a prover to prove to a verifier the knowledge of the representation of an elements y, to the bases $g_1, \ldots, g_n$. Using the notation in [25], the protocol is denoted by:

$$PK\{(\alpha_1, \ldots, \alpha_n) : y = g_1^{\alpha_1} \ldots g_n^{\alpha_n}\}.$$

A prover $\mathcal{P}$ knowing $x_1, \ldots, x_n \in \mathbb{Z}$ such that $y_i = g_i^{x_i}$ for all $i = 1, \ldots, n$ can prove to a verifier $\mathcal{V}$ his/her knowledge as follows.

- (Commit.) $\mathcal{P}$ chooses $r_i \in_R \mathbb{Z}_{(2^{\ell_i} q)^\epsilon}$ and computes $t_i \leftarrow g_i^{r_i}$ for all $i = 1, \ldots, n$. $\mathcal{P}$ sends $(t_1, \ldots, t_n)$ to $\mathcal{V}$.

- (Challenge.) $\mathcal{V}$ chooses $c \in_R \mathbb{Z}_q$ and sends it to $\mathcal{P}$.
- (Response.) $\mathcal{P}$ computes, for all $i = 1, \ldots, n$, $s_i \leftarrow r_i - cx_i$ (in $\mathbb{Z}$). $\mathcal{P}$ sends $(s_1, \ldots, s_n)$ to $\mathcal{V}$.

$\mathcal{P}$ verifies by checking, if $t_1 \ldots t_n \stackrel{?}{=} g_1^{s_1} \ldots g_n^{s_n} y^c$.

Theorem 2 If the Strong RSA assumption holds, the protocol stated in Sec. 2.5.3 is an HVZK PoK protocol.

Proof 2 The proof is a straightforward extension from the proof of in [32] and [18]. □

As noted before, the protocol can be turned into a signature scheme by replacing the challenge by the hash of the commitment together with the message M to be signed: $c \leftarrow H(g_1 || \ldots || g_n || y || t_1 \ldots t_n || M)$. In this case, the signature is $(c, s_1, \ldots, s_n)$ and the verification becomes:

$$c \stackrel{?}{=} H(g_1 || \ldots || g_n || y || g_1^{s_1} \ldots g_n^{s_n} y^c || M).$$

Following [25], we denote this signature scheme by:

$$SPK\{(\alpha_1, \ldots, \alpha_n) : y = g_1^{\alpha_1} \ldots g_n^{\alpha_n}\}(M).$$

2.5.4 Proving the Knowledge of d Out of n Equalities of Discrete Logarithms

This protocol is constructed using the techniques described in [37], by combining the PoK for discrete logarithm in [18] and the secret sharing scheme due to Shamir [92]. This allows a prover to prove to a verifier his/her knowledge of some d out of n integers $x_1, \ldots, x_n$, where $x_i = \log_{g_i} y_i = \log_{h_i} v_i$ for all

$i = 1, \ldots, n$. The protocol is denoted by:

$$PK\left\{(\alpha_1, \ldots, \alpha_n) : \bigvee_{\mathcal{J} \in \wp_d([1,n])} \left(\bigwedge_{i \in \mathcal{J}} y_i = g_i^{\alpha_i} \wedge v_i = h_i^{\alpha_i} \right) \right\}.$$

A prover $\mathcal{P}$ knowing, for all $i \in \mathcal{I}$, $x_i \in \mathbb{Z}$ such that $y_i = g_i^{x_i}$ and $v_i = h_i^{x_i}$, where $\mathcal{I}$ is some subset of $\mathcal{N}$ such that $|\mathcal{I}| = d$, can prove his/her knowledge to a verifier $\mathcal{P}$ as follows.

- (Commit.) $\mathcal{P}$ does the following: For $i \in \mathcal{N} \setminus \mathcal{I}$, select $c_i \overset{R}{\leftarrow} \mathbb{Z}_q$. For all $i \in \mathcal{N}$, select $r_i \overset{R}{\leftarrow} \mathbb{Z}_{(2^{\ell_i} q)^\epsilon}$. Compute

$$t_i \leftarrow \begin{cases} g_i^{r_i}, & i \in \mathcal{I}; \\ g_i^{r_i} y_i^{c_i}, & i \in \mathcal{N} \setminus \mathcal{I}, \end{cases} \quad \text{and } T_i \leftarrow \begin{cases} h_i^{r_i}, & i \in \mathcal{I}; \\ h_i^{r_i} v_i^{c_i}, & i \in \mathcal{N} \setminus \mathcal{I}. \end{cases}$$

  $\mathcal{P}$ sends $(t_1, \ldots, t_n, T_1, \ldots, T_n)$ to $\mathcal{V}$.

- (Challenge.) $\mathcal{V}$ chooses $c \in_R \mathbb{Z}_q$ and sends it to $\mathcal{P}$.

- (Response.) $\mathcal{P}$ does the following: Compute a polynomial f of degree $\leq n - d$ over $\mathbb{Z}_q$ such that $f(0) = c$ and $f(i) = c_i$ for all $i \in \mathcal{N} \setminus \mathcal{I}$. Compute $c_i \leftarrow f(i)$ for all $i \in \mathcal{I}$. Set

$$s_i \leftarrow \begin{cases} r_i - c_i x_i, & i \in \mathcal{I}; \\ r_i, & i \in \mathcal{N} \setminus \mathcal{I}. \end{cases}$$

  $\mathcal{P}$ sends $(f, s_1, \ldots, s_n)$ to $\mathcal{V}$.

$\mathcal{P}$ verifies by checking if (1) f is a polynomial of degree $\leq n - d$ over $\mathbb{Z}_q$, (2) $f(0) \overset{?}{=} c$, and (3) $t_i \overset{?}{=} y_i^{f(i)} g_i^{s_i}$ and $T_i \overset{?}{=} v_i^{f(i)} h_i^{s_i}$, for all $i = 1, \ldots, n$.

Theorem 3 If the Strong RSA assumption holds, the protocol stated in Sec. 2.5.4 is an HVZK PoK protocol.

(*Proof Sketch*) To prove the theorem, it suffices to show that the protocol is correct, sound and statistical HVZK.

- (Correctness.) Straightforward.
- (Soundness.) It suffices to show how a witness can be extracted if given two valid protocol conversations with the same commitment but different challenges. Denoting the two conversation transcripts by

$$\langle (t_1, \ldots, t_n, T_1, \ldots, T_n), (c), (f, s_1, \ldots, s_n) \rangle \text{ and}$$

$$\langle (t_1, \ldots, t_n, T_1, \ldots, T_n), (c'), (f', s'_1, \ldots, s'_n) \rangle,$$

  we have $c \neq c'$ and thus $f(0) \neq f'(0)$. As the degrees of f and f' are at most $n-d$, there are at least d distinct values $\pi_1, \ldots, \pi_d \in \{1, \ldots, n\}$ such that $f(\pi_i) \neq f'(\pi_i)$ for all $i = 1, \ldots, d$. Using arguments in [18], $f(\pi) - f'(\pi)$ divides $s'_\pi - s_\pi$ and therefore an integer $\hat{x}$ such that $y_\pi = g_\pi^{\hat{x}\pi}$ and $v_\pi = h_\pi^{\hat{x}\pi}$ can be computed as: $\hat{x}_\pi \leftarrow (s_\pi - s'_\pi)/(f'(\pi) - f(\pi))$.

  Hence a witness $(\hat{x}_{\pi_1}, \ldots, \hat{x}_{\pi_d})$ can be computed from two such transcripts.

- (Statistical HVZK.) To simulate a transcript, a simulator $\mathcal{S}$ first chooses uniformly at random a polynomial f' of degree $n-d$ over $\mathbb{Z}_q$. For all $i = 1, \ldots, n$, $\mathcal{S}$ picks uniformly at random $s'_i \in_R \mathbb{Z}_{(2^{\ell_i} q)^\epsilon}$ and computes $t'_i \leftarrow g_i^{s'_i} y_i^{f'(i)}$. The simulated transcript is:

$$\langle (t'_1, \ldots, t'_n, T'_1, \ldots, T'_n), (f'(0)), (f', s'_1, \ldots, s'_n) \rangle.$$

  To prove that the simulation is statistical indistinguishable from real protocol conservations, one should consider, for each $i = 1, \ldots, n$, the probability distribution $P_{S_i}(s_i)$ of the responses of the prover and the probability distribution $P_{S'_i}(s'_i)$ according to which $\mathcal{S}$ chooses s'_i. The

statistical distance between the two distributions can be computed to be at most: $2(2^{\ell_i})(q-1)/(2^{\ell_i}q)^{\epsilon} \leq 2/(2^{\ell_i}q)^{\epsilon-1}$. The result follows.

□

The protocol can be turned into a signature scheme by replacing the challenge by the hash of the commitment together with the message M to be signed:

$$c \leftarrow H((g_1, y_1, h_1, v_1)|| \ldots ||(g_n, y_n, h_n, y_n)||t_1|| \ldots ||t_n||T_1|| \ldots ||T_n||M).$$

In this case, the signature is $(f, s_1, \ldots, s_n)$ and step (3) of the verification becomes:

$$\begin{aligned} c \stackrel{?}{=} H(\quad & (g_1, y_1, h_1, v_1)|| \ldots ||(g_n, y_n, h_n, y_n)|| \\ & y_1^{c_1} g_1^{s_1}|| \ldots ||y_n^{c_n} g_n^{s_n}||v_1^{c_1} h_1^{s_1}|| \ldots ||v_n^{c_n} h_n^{s_n}||M). \end{aligned}$$

We denote this signature scheme by:

$$SPK\left\{(\alpha_1, \ldots, \alpha_n) : \bigvee_{\mathcal{J} \in \wp_d([1,n])} \left(\bigwedge_{i \in \mathcal{J}} y_i = g_i^{\alpha_i} \wedge v_i = h_i^{\alpha_i} \right) \right\}(M).$$

2.6 Conclusion

In this chapter, we have provided all the necessary background and foundations of cryptography that will be used in the subsequent chapters. We have given an introduction to the topics of complexity theories, algebra, number theory. We have reviewed various cryptographic primitives including encryption, digital signatures, etc. Finally we have elaborated on zero-knowledge proof of knowledge protocols.

Chapter 3

Related Works

In this chapter, we survey the literature on works related to our thesis. They serve as a good tutorial on various security goals and notions, current state-of-art technology, similarities and differences among schemes. We hope that after reading this chapter, the readers can better understand the incentives that drive the writing of this thesis, and at the same time better evaluate the contribution of this thesis.

3.1 Introduction

In this section we introduce *group-oriented* signature schemes. In signature schemes that are not group-oriented, we have each entity represented by a single user. For example, one user signs in a conventional signature scheme, one user signs and one user is designated to verify in a designated-verifier signature scheme [31, 60], etc. In group-oriented signature schemes, however, more than one user is representing an entity. For example, more than one user jointly plays the role of the signer in a threshold signature scheme [41], multi-signature scheme [59], etc.

In some group-oriented signature schemes, *anonymity* of the signers and/or verifiers is not a concern and is thus not guaranteed. These schemes are used when the signers feel okay about letting others know that they have helped

in generating a signature. These schemes may also be employed when the verifiers would like to know who exactly were involved in the signing process. On the other hand, some group-oriented signatures are not *spontaneous*, and require a proprietary setup stage before a signature can be signed even when a PKI already exists among the users. Spontaneity is a nice property for group-oriented signature schemes because it gets rid of the need of a powerful and/or trusted group manager. Among some of the various group-oriented signature schemes, threshold signatures and group signatures are not spontaneous, multi-signatures, aggregated signatures and designated verifiers signatures are not anonymous.

In the next two subsections, we give a review on both group-oriented signature schemes without spontaneity and/or anonymity and spontaneous anonymous group-oriented (SAG) signature schemes.

3.2 Group-Oriented Signatures without Spontaneity and/or Anonymity

We briefly review group-oriented signature schemes without spontaneity and/or anonymity in the following.

THRESHOLD SIGNATURES. The secret key is distributed among n parties in a threshold signature scheme either with the help of a trusted dealer or by running an interactive protocol among all parties. To sign a message M any t (but not less) parties use their shares of the secret and run an signature generation protocol. A secure threshold signature scheme must make existential forgery impossible even if some $t-1$ parties have been corrupted. Non-interactiveness means the participating signers need not communicate in the process of signature generation. If a threshold signature scheme is robust, a signature can still be generated if number of the signers acting adversarily is within a certain

limit. Research in the literature: [41, 39, 49, 57, 50, 87, 54, 93, 71, 11, 102, 35].

MULTI-SIGNATURES. In a multi-signature scheme, any subgroup of a group of players jointly sign a document such that a verifier is convinced that each member of the subgroup participated in signing. The difference between multi-signatures and threshold signatures is the following: multi-signatures prove that each member of the stated subgroup signed the message while threshold signatures prove that some subgroup of sufficient size signed the message. Research in the literature: [59, 84, 57, 66, 58, 85, 74, 11, 101].

VERIFIER DESIGNATED SIGNATURES. Verifier designated signature (VDS) schemes, independently proposed in [31] and [60] are such that a signature can only be verified by the verifier who is designated. Recently, Desmedt suggested in [40] to generalize the number of verifiers to an arbitrary number. The generalized scheme becomes obviously a group-oriented scheme. Laguillaumie, et al. [65] proposed a VDS scheme that hides the signer's identity among a group of possible signers, by incorporating the ring structure [91]. The same authors also proposed in [64] a signer-anonymous multi-designated verifier signature scheme. Papers in this topic include: [31, 60, 65, 64].

Note that the identities of designated verifiers are known to the public in all existing multi-designated verifier signature schemes, even in those in which signers are anonymous.

GROUP SIGNATURES. Introduced by Chaum in [34], group signatures allow a member to sign messages anonymously on behalf of his group. The group manager is responsible to form the group and assign to the members the ability to sign. However, in the case of a dispute, the identity of a signature's originator can be revealed (only) by a designated entity. The first efficient and provably secure group signature scheme was due to [3]. The requirement of group setup by the group manager in group signature schemes prevents them from being spontaneous. There has been fruitful research since then: [19, 21, 13, 22, 82, 78, 75, 53, 8].

Teranishi, et al. [95] proposed an authentication scheme in which users can be authenticated anonymously so long as times that they are authenticated is within an allowable number. In the scheme, no one is capable of identifying users who have been authenticated within the allowable number; but anyone can trace dishonest users who have been authenticated beyond the allowable number. When one regards signature schemes as the non-interactive version of authentication, [95] can actually be regarded as a group signature with linkability [76, 77, 68] or a credential system [19, 19, 20, 22, 70] with multi but restricted number of shows.

3.3 SAG Signatures

As discussed above, SAG signatures are group-oriented signatures that are *both* spontaneous and anonymous (anonymity usually refers to that of the signers). They are thus sometimes regarded as group signatures with spontaneity: there is no group setup (either by a group manager or through interactions among users).

RING SIGNATURES. The first SAG signature scheme is due to Rivest, et al. [91] which has a structure of a ring and was thus given the name *ring signatures*. The notion "ring signatures" is, however, sometimes exploited a little bit to mean SAG signatures in general, and has nothing to do with the structure of the construction itself. We also adopt this nomenclature in this paper and refer to SAG signatures as ring signatures from time to time. On the other hand, when we say an SAG signature scheme is of ring-type, we mean the scheme has a structure of a ring, similar to that in [91].

The construction of SAG signatures actually dates back to [37] in which *partial proof of knowledge* was introduced. Such a proof protocol allows a prover to prove to a verifier his knowledge of a witness behind certain relation among a set of relations. Using the Fiat-Shamir heuristics [46], the protocol

can be transformed into a signature scheme and results in a group signature scheme (secure in the Random Oracle model [7]) that enjoys spontaneity as well as anonymity. SAG signature schemes constructed this way are often referred to as of CDS-type. Note that it is trivial to extend a CDS-type, but not a ring-type, SAG signature scheme into a threshold setting with the same order computational and storage efficiency.

Most of the SAG signature schemes in the literature today either adopt a ring structure or a CDS structure in their construction. However there are exceptions. For example, Boneh, et al.'s ring signature scheme [14] makes use of bilinearity of pairings to achieve its goal.

Abe, et al. [1] gave a construction of ring signatures, with *separability* taken into account. Separability was introduced in [63] and diversified in [23] to describe the users' ability to choose their own cryptographic primitive and system parameters. Consult [23] for various levels of separability. Separability is of vital importance in SAG cryptography as there is no group manager or trusted third party to coordinate the choice of primitives and system parameters for each user. For instance, a SAG signature scheme that is not separable is not practical at all as it is unlikely to have all group members using the same primitive, system parameters and security parameters.

Dodis, et al. [43] proposed an anonymous identification scheme that allows participants from a user population to form ad hoc groups, and then prove membership anonymously in such groups. Using the Fiat-Shamir transform [46], a signer-ambiguous SAG signature scheme can be obtained. It is the first SAG signature scheme that has a constant signature size (independent of the size of the group).

Regarding the security of SAG signature schemes, a secure SAG signature scheme must be unforgeable and anonymous. Roughly speaking, unforgeability means a valid signature can only be generated by a group member while

anonymity means no one can decide the origin of a signature better than random guessing. Papers that concern about security model for SAG signature schemes include: [91, 1, 69].

THRESHOLD RING SIGNATURES. Threshold cryptography [41] allows n parties to share the ability to perform a cryptographic operation (e.g., creating a digital signature). Any d parties can perform the operation jointly, whereas it is infeasible for at most $d-1$ to do so. In a (d, n)-threshold ring signature scheme, the generation of a ring signature for a group of n members requires the involvement of at least d members/signers, and yet the signature reveals nothing about the identities of the signers. A threshold ring signature scheme effectively proves that a certain minimum number of users of a certain group must have actually collaborated to produce the signature, while hiding the precise membership of the subgroup.

Bresson, et al. [16] was the first to study *ad-hoc ring signatures*, which is a generalization of threshold ring signatures. An ad-hoc group is a list of users, including certified public keys, accompanied by a list of subsets of these users, called the acceptable subsets. This second list may be optionally replaced with a predicate defining exactly which subsets are acceptable. Ad-hoc ring signatures prove that the signing members all belong to at least one acceptable subset. Other threshold ring signature schemes include: [100, 67].

BLIND RING SIGNATURES. While ring signatures protect the anonymity of the signer, the blindness in blind ring signatures blinds the messages to be signed against the signer. The first blind ring signatures were introduced by Chan, et al. [28]. Based on essentially any major blind signature scheme, they constructed ring-type 1-out-of-n blind ring signatures and CDS-type t-out-of-n blind ring signatures. The blindness of the various resulted blind ring signature schemes depends on that of their respective underlying blind signature schemes.

LINKABLE RING SIGNATURES. Liu, et al. [68] gave the first linkable ring

signatures. Linkability in ring signatures allow anyone to determine if they are signed by the same group member (i.e. they are *linked*). If a user signs only once on behalf of a group, he still enjoys anonymity similar to that in conventional ring signature schemes. If the user signs multiple times, anyone can tell that these signatures have been generated by the same group member.

Applications include leaking sequences of secrets, e-voting [68], offline anonymous electronic cash systems, direct anonymous attestations [97] and restricted one-show credential systems [70].

DENIABLE RING AUTHENTICATION. The notion was introduced by Naor [79]. In a deniable ring authentication, it is possible to convince a verifier that a member of an ad hoc collection of participants is authenticating a message without revealing which one and the verifier cannot convince any third party that the message was indeed authenticated.

Susilo, et al. later proposed a non-interactive version [94] which can get rid of inefficient implementation of anonymous channel. They also presented in the same paper an extension of this idea to allow a non-interactive deniable ring to threshold ring authentication. In this scenario, the signature can convince a group of verifiers, but the verifiers cannot convince any other third party about this fact, because any collusion of t verifiers can always generate a valid message-signature pair.

3.4 Conclusion

In this chapter, we have surveyed the literature on works related to our thesis. They serve as a good tutorial on various security goals and notions, current state-of-art technology, similarities and differences among schemes. We hope that after reading this chapter, the readers can better understand the incentives that drive the writing of this thesis, and at the same time better evaluate the contribution of this thesis.

Chapter 4

Linkable Ring Signatures

In this chapter, we investigate in depth one extension of ring signatures, namely the linkable ring signatures. We first give an introduction of linkable ring signatures and the significance of linkability in ring signatures. Then we introduce notions novel to linkable ring signatures, together with discussion on their importance and impact. Finally we give a fully developed security model that captures the security requirements of linkable ring signature schemes under various possible adversarial attacks.

4.1 Introduction

Linkable ring signatures [68] are ring signatures, but with added linkability: such signatures allow anyone to determine if they are signed by the same group member (i.e. they are *linked*). If a user signs only once on behalf of a group, he still enjoys anonymity similar to that in conventional ring signature schemes. If the user signs multiple times, anyone can tell that these signatures have been generated by the same group member. Applications include leaking sequences of secrets and e-voting [68]. Concepts similar to linkability also appeared in one-show credentials [19], linkable group signatures [76, 77], and DAA [17].

The most crucial significance of linkability in ring signatures is to take away their perfect anonymity and thus making them suitable for new applications.

Ring signatures provide anonymity in a very strong sense: signatures signed by different users are statistically indistinguishable from each other, meaning even adversaries with infinite power cannot tell who actually signed. The only way to reveal the identity is to have the actual signer cooperate by either revealing his secret or going through some (non-)interactive proof protocols. While this is excellent to provide users with perfect anonymity, such a scheme is not suitable when absolute anonymity is a threat. In fact, in not many situations is perfect anonymity required. On the contrary, there are lots of applications in which information about the identity of the actual signer can be obtained. Group signatures is definitely a clear example: anonymity is preserved except to the revocation manager, who can open a signature in case of a dispute or under a court order. Linkability can be thought of a mechanism that leak partial information about signer's identity if a predefined criterion is met. When compared to group signatures, linkable ring signatures offer a distinct feature that is possibly advantageous in certain applications: In group signatures, the revocation manager is empowered to open any signatures; while in linkable ring signatures, no one has a clue to a signature's originality, provided that the predefined criterion (namely double-signing) is not met.

Our Contributions in this Chapter

- We introduce new notions to linkable ring signatures: (1) *Non-accusatory linkability* only detects the presence of two "linked" signatures, while *accusatory linkability* additionally outputs the identity of the suspected "double-signer". (2) *Non-slanderability* means no coalition can generate signatures accusatorily linked to a targeted victim. (3) We clarify the meaning of linkability in linkable threshold ring signature schemes.

- We formally define the "*event-oriented*" linking criterion. Under such linkability, one can tell if two signatures are linked if and only if they are

signed for the same event, despite the fact that they may be signed on behalf of different groups.

- We give a fully developed security model that captures the security requirements of generic linkable ring signature schemes under various possible adversarial attacks.

4.2 New Notions

4.2.1 Accusatory Linking

In the first linkable ring signature scheme due to [68], the linking algorithm can tell if two input signatures are linked or not, meaning whether or not they are signed by the same member of a group. The boolean answer given by the linking algorithm reveals nothing about the actual identity of the signer, even if two signatures are linked. Linkability in a ring signature scheme is good because it serves as a mechanism to hinder "double-signing". A signer, knowing "double-signing" is detectable, loses interest to so do. A good example to illustrate the idea is when linkable ring signatures are used for electronic voting. A linkable ring signature is signed on a to-be-casted ballot to authenticate its eligibility. An adversarial voter may want to vote twice or more to unfairly put his favorable candidate into a more advantageous position. If the voting authority throws away all the votes with linked signatures, the adversary has no longer the incentive to "double-vote", because not only can't he vote twice or more, his originally legitimate ballot will be discarded as well. We call linking without being able to figure out the identity of the double-signer *Non-accusatory Linking.*

However, there are occasions when non-accusatory linking is not good enough to prevent double-signing. This may happen when (1) there is nothing to lose for a signer to double-sign, or (2) linkability test is done after, instead

of concurrently with, the verification of a signature. The consequences of the former case is easy to see. Because the signer has nothing to lose, he is happy to sign thousands of times, in the hope of two or more signatures are undetected as unlinked. He may also desperately sign a countless number of times to keep the linkability checking authority busy, which results in a denial of service attack.

In the latter case, linkability test is done after the verification of a signature. The best example to explain the potential problem raised is perhaps offline anonymous eletronic cash systems that make use of linkable ring signatures. In such systems, each signature represents a coin to be spent. Double-signing has in this case the physical meaning of double-spending, i.e. spending the same coin twice. Since linking is done only when a coin is deposited at the bank by the merchant, the merchant is only able to verify its validity, but has no way to know if the coin is being double-spent. A double spender, having benefited by exchanging the double-spent coins with the goods or service from the merchant, does not care that the coins will be linked at a later time. Switching the whole system into an online one is a trivial solution because linkability can be tested during payment. But if being online is not what we want, we need to figure out the identity of the double signer from two linked signatures, so as to punish him for double signing. When a linkable ring signature scheme is able to identify the double signer from two linked signatures, We call it *Accusatory Linking.*

4.2.2 Non-slanderability

Roughly speaking, non-slanderability means no one can produce a valid signature that is linked to a given signature, except the one who actually signed that signature. The full definition of non-slanderability is given in the security model later in this chapter.

Non-slanderability is important for linkable ring signature schemes because schemes that are slanderable can cause so much problems and make it unsuitable to be employed for practical use. Again we demonstrate the idea by the example of electronic voting. In electronic voting, linked signatures are dropped. Therefore, in a linkable ring signature scheme is slanderable, even if a honest voter never double vote, it is possible that his vote is dropped because someone is able to produce a valid signature that is linked to the honest voter. As a result, an adversary can void any votes as he wish. He can practically ruin the whole voting event or control the winner in the event.

4.2.3 Linkability in Threshold Ring Signatures

While it is trivial what linkability means in conventional ring signatures, linkability in threshold ring signatures requires a more precise definition. In particular, there are at least two possibilities that we define as being *coalition-linkable* and *individual-linkable* as follows.

Definition 25 (Coalition-linkability.) *A linkable threshold ring signature scheme is coalition-linkable if two signatures are linked if and only if they are signed by exactly the same set of signers.*

Definition 26 (Individual-linkability.) *A linkable threshold ring signature scheme is individual-linkable if two signatures are linked if and only if they involve a common signer.*

In a linkable threshold ring signature scheme that is coalition-linkable, users are able to sign multiple times without their signatures being linked, as long as they are not collaborating with exactly the same set of signers again. However, in a scheme that is individual-linkable, no matter who other collaborating signers are a user signing more than once will have the signatures linked. The scheme we present in this paper falls into the later category.

4.2.4 Event-Oriented Linking

In the first linkable ring signature scheme [68], one can tell if two ring signatures are linked or not if and only if they are signed on behalf of the same group of members. We call this "*group-oriented*" linkability. We formulate a new linking criterion that we call "*event-oriented*" linkability in which one can tell if two signatures are linked if and only if they are signed for the same event, despite that they may be signed on behalf of different groups.

Event-oriented linkable ring signatures are comparatively more flexible in application. We illustrate by the following two examples:

Example 1 group settings keep changing frequently in ad-hoc groups and most of the ring signatures are signed on behalf of different groups, thus rendering group-oriented linkability virtually useless.

Example 2 The CEOs of a company vote for business decisions. Using linkable ring signatures, they can vote anonymously by ring-signing their votes. However, as the group is fixed throughout the polls, votes among polls can be linked by anybody and information can be derived which means anonymity is in jeopardy. This can be prevented when an event-oriented scheme is used.

In deploying a linkable (threshold) ring signature scheme that supports event-oriented linking, event-ids should be chosen with great care. Here we give two examples.

Example 3 When the scheme is used to leak sequences of secrets, the "whistle-blower" should choose an event-id that has never been used before when leaking the first secret and then stick to using the same later on. This guarantees the sequence of secrets cannot be linked to other sequences.

Example 4 When the scheme is used in electronic voting, it is usually the voting organizer (e.g. the government) who decides on an event-id. Each

eligible voter should therefore, before they cast a vote, make sure that the event-id has not been used in any previous voting event, so as to secure the intended privacy.

4.3 Security Model

We give our security model and define relevant security notions.

4.3.1 Syntax

A *Linkable Threshold Ring Signature* (LTRS) scheme, is a tuple of five algorithms (Init, Key-Gen, Sign, Verify and Link).

- $\mathsf{param} \leftarrow \mathsf{Init}(1^\lambda)$, a PPT *Initialization* algorithm which, on input a security parameter $\lambda \in \mathbb{N}$, outputs the set of system's parameters param which also includes 1^λ.

- $(sk_i, pk_i) \leftarrow \mathsf{Key\text{-}Gen}(\mathsf{param}, 1^{\lambda_i})$, a PPT *Key Generation* algorithm which, on input the system's parameters param and a further security parameter $\lambda_i \in \mathbb{N}$ such that $\lambda_i \geq \lambda$, outputs a secret/public key pair (sk_i, pk_i). We denote by $\mathcal{SK}$ and $\mathcal{PK}$ the domains of possible secret keys and public keys, resp. When we say that a public key corresponds to a secret key or vice versa, we mean that the secret/public key pair is an output of Key-Gen.

- $\sigma \leftarrow \mathsf{Sign}(\mathsf{param}, e, n, d, \mathcal{Y}, M, \mathcal{X})$, a PPT *Signing* algorithm which, on input the set of system's parameters param, an event-id $e \in \{0,1\}^*$, a group size $n \in \mathbb{N}$ with size polynomial in λ, a threshold $d \in [1, n]$, a set $\mathcal{Y}$ of n public keys in $\mathcal{PK}$, a message $M \in \{0,1\}^*$, and a set $\mathcal{X}$ of d private keys in $\mathcal{SK}$ whose corresponding public keys are contained

in $\mathcal{Y}$, produces a signature σ. We denote by Σ the domain of possible signatures. For convenience define the augmented signature σ_{aux} as $(\mathsf{param}, e, n, d, \mathcal{Y}, M, \sigma)$.

- $1/0 \leftarrow \mathsf{Verify}(\mathsf{param}, e, n, d, \mathcal{Y}, M, \sigma)$, a polynomial-time *Verification* algorithm which, on input the set of system's parameters param, an event-id $e \in \{0,1\}^*$, a group size $n \in \mathbb{N}$ with size polynomial in λ, a threshold $d \in [1, n]$, a set $\mathcal{Y}$ of n public keys in $\mathcal{PK}$, a message $M \in \{0,1\}^*$ and a signature $\sigma \in \Sigma$, returns 1 or 0 for `accept` or `reject` respectively. If the algorithm returns `accept`, the message-signature pair (M, σ) is said to be *valid* (w.r.t. $(\mathsf{param}, e, n, d, \mathcal{Y})$).

- $1/0 \leftarrow \mathsf{Link}(\sigma_{aux}^{(0)}, \sigma_{aux}^{(1)})$, a polynomial-time *Linking* algorithm which, upon input two valid augmented signatures with respect to the same set of system's parameters param and event-id e, outputs 1 or 0 for `linked` or `unlinked`, resp. If the scheme's linkability is *accusatory*, the algorithm additionally outputs the public key pk_{sus} of the suspected "double-signer" in case of `linked`.

The syntax for a Linkable Ring Signature (LRS) is a straightforward special case of the above when the threshold value d is always 1. For schemes that do not support events, one may simply assume they support a single event-id. Therefore, the above syntax incorporates linkable ring signature schemes no matter they support thresholding and/or events or not.

Correctness

LTRS schemes must satisfy:

- (Verification Correctness.) Signatures signed according to specification are accepted during verification, with overwhelming probability.

- (Linking Correctness.) Two signatures signed for the same event according to specification are `linked` with overwhelming probability if the two signatures share a common signer. On the other hand, two signatures signed for the same event according to specification are `unlinked` with overwhelming probability if the two signatures do NOT share a common signer. In case linkability is accusatory, the suspect output by the algorithm Link is, with overwhelming probability, the common signer when the two input signatures are `linked`.

4.3.2 Notions of Security

The security of a LTRS scheme has four aspects: unforgeability, linkable anonymity, linkability, and non-slanderability. Before giving their definition, we consider the following oracles which together model the ability of the adversaries in breaking the security of the scheme.

- The *Joining Oracle* $\mathcal{JO}$. Upon request, it adds a new user to the system and then returns the public key $pk \in \mathcal{PK}$ of that new user.

- The *Corruption Oracle* $\mathcal{CO}$. On input a public key pk_i that is a query output of $\mathcal{JO}$, it returns the corresponding secret key $sk_i \in \mathcal{SK}$.

- The *Signing Oracle* $\mathcal{SO}$. On input an event-id $e \in \{0,1\}^*$, a group size $n \in \mathbb{N}$ of size polynomial in the security parameter λ, a threshold $d \in [1,n]$, a set $\mathcal{Y}$ of n public keys that are query outputs of $\mathcal{JO}$, a message $M \in \{0,1\}^*$, and a set $\mathcal{V} \subseteq \mathcal{Y}$ of size d, it returns a valid signature σ signed by the users with pubic keys in $\mathcal{V}$.

Remark: An alternative approach to specify the $\mathcal{SO}$ is to exclude the signer set $\mathcal{V}$ from the input and have $\mathcal{SO}$ select it according to suitable random distribution. We do not pursue that alternative further.

Unforgeability

Definition 27 (Game Unf.) *Unforgeability for LTRS schemes is defined in the following game between the Simulator $\mathcal{S}$ and the Adversary $\mathcal{A}$ in which $\mathcal{A}$ is given access to oracles $\mathcal{JO}$, $\mathcal{CO}$ and $\mathcal{SO}$:*

1. (Initialization Phase.) *$\mathcal{S}$ generates and gives $\mathcal{A}$ the system's parameters* param.
2. (Probing Phase.) *$\mathcal{A}$ may query the oracles according to any adaptive strategy.*
3. (Challenge Phase.) *$\mathcal{A}$ gives $\mathcal{S}$ an event-id $e \in \{0,1\}^*$, a group size $n \in \mathbb{N}$ of size polynomial in λ, a threshold $d \in [1,n]$, a set $\mathcal{Y}$ of n public keys in $\mathcal{PK}$, a message $M \in \{0,1\}^*$ and a signature $\sigma \in \Sigma$.*

In the above game, $\mathcal{A}$ wins if:

1. $\mathsf{Verify}(\mathsf{param}, e, n, d, \mathcal{Y}, M, \sigma) = 1$,
2. all of the public keys in $\mathcal{Y}$ are query outputs of $\mathcal{JO}$,
3. at most $(d-1)$ of the public keys in $\mathcal{Y}$ have been input to $\mathcal{CO}$, and
4. σ is not a query output of $\mathcal{SO}$ on any input containing M.

We denote by $\mathbf{Adv}^{unf}_{\mathcal{A}}(\lambda)$ the probability of $\mathcal{A}$ winning the game.

Definition 28 (Unforgeability.) *An LRS scheme is unforgeable if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}^{unf}_{\mathcal{A}}(\lambda)$ is negligible.*

Linkable Anonymity

Definition 29 (Game L-Anon.) *Linkable anonymity for LTRS schemes is defined in the following game between the Simulator $\mathcal{S}$ and the Adversary $\mathcal{A}$ in which $\mathcal{A}$ is given access to oracles $\mathcal{JO}$, $\mathcal{CO}$ and $\mathcal{SO}$:*

1. (Initialization Phase.) $\mathcal{S}$ *generates and gives* $\mathcal{A}$ *the system's parameters* param.

2. (Probing Phase I.) $\mathcal{A}$ *may query the oracles according to any adaptive strategy.*

3. (Challenge Phase.) $\mathcal{A}$ *gives* $\mathcal{S}$ *an event-id* e^*, *a group size* $n^* \in \mathbb{N}$ *of size polynomial in* λ, *a threshold* $d^* \in [1, n^*]$, *a message* M^*, *a set* $\mathcal{Y}^*$ *of* n^* *public keys that are query outputs of* $\mathcal{JO}$, *a set* $\mathcal{V}^* \subseteq \mathcal{Y}^*$ *of size* d^*, *a public key* $y^* \in \mathcal{V}^*$ *that has never been input to* $\mathcal{CO}$ *or included in the insider set* $\mathcal{V}$ *in any query to* $\mathcal{SO}$. *Let* x^* *be the secret key corresponding to* y^* *and* $\mathcal{U}^*$ *be the set of secret keys corresponding to the public keys contained in* $\mathcal{V}^*$.

   $\mathcal{S}$ *then selects* $b \in_R \{0,1\}$. *Case* $b = 0$: $\mathcal{S}$ *computes* σ^* *by executing the algorithm* Sign *on the input tuple* $(\mathsf{param}, e^*, n^*, d^*, \mathcal{Y}^*, \mathcal{U}^*, M^*)$. *Case* $b = 1$: $\mathcal{S}$ *computes* $\sigma^* = \mathcal{SO}$ $(e^*, n^*, d^*, \mathcal{Y}^*, \mathcal{V}^*, \mathcal{X}^*, M^*)$.

   $\mathcal{S}$ *sends* σ^* *to* $\mathcal{A}$.

4. (Probing Phase II.) $\mathcal{A}$ *queries the oracles adaptively, except that* y^* *cannot be queried to* $\mathcal{CO}$ *or included in the insider set* $\mathcal{V}$ *of any query to* $\mathcal{SO}$.

5. (End Game.) $\mathcal{A}$ *delivers an estimate* $\hat{b} \in \{0,1\}$ *of* b.

In the above game, $\mathcal{A}$ wins if $\hat{b} = b$. We denote by $\mathbf{Adv}_{\mathcal{A}}^{L-Anon}(\lambda)$ the probability of $\mathcal{A}$ winning the game over one-half.

Definition 30 (Linkable Anonymity.) *An LTRS scheme is linkably anonymous if for any PPT adversary* $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{L-Anon}(\lambda)$ *is negligible.*

Remark: Linkable anonymity is a form of computational zero-knowledge: the attacker cannot computationally distinguish the real world from the simulated world. Note that the anonymity notions in [6, 9, 61] appear to be

also computational zero-knowledge. Our attacker model is not a fully active attacker: queries relevant to the gauntlet public key, y_g, are ruled out. The anonymity in [68] is also with respect to the above model. We note that [6], p.623, argued that anonymity and linkability cannot coexist in their security model.

Linkability

Definition 31 (Game Link.) *Linkability for LTRS schemes is defined in the following game between the Simulator $\mathcal{S}$ and the Adversary $\mathcal{A}$ in which $\mathcal{A}$ is given access to oracles $\mathcal{JO}$, $\mathcal{CO}$ and $\mathcal{SO}$:*

1. (Initialization Phase.) *$\mathcal{S}$ generates and gives $\mathcal{A}$ the system's parameters* param.
2. (Probing Phase.) *$\mathcal{A}$ may query the oracles according to any adaptive strategy.*
3. (Challenge Phase.) *$\mathcal{A}$ gives $\mathcal{S}$ an event-id $e \in \{0,1\}^*$, group sizes $n^{(1)}, n^{(2)} \in \mathbb{N}$, thresholds $d^{(1)} \in [1, n^{(1)}], d^{(2)} \in [1, n^{(2)}]$, sets $\mathcal{Y}^{(1)}$ and $\mathcal{Y}^{(2)}$ of public keys that are query outputs of $\mathcal{JO}$ of sizes $n^{(1)}$ and $n^{(2)}$ respectively, messages $M^{(1)}, M^{(2)} \in \{0,1\}^*$ and signatures $\sigma^{(1)}, \sigma^{(2)} \in \Sigma$.*

In the above game, $\mathcal{A}$ wins if

1. *all public keys in $\mathcal{Y}^{(1)} \cup \mathcal{Y}^{(2)}$ are query outputs of $\mathcal{JO}$,*
2. *Verify*$(\mathsf{param}, e, n^{(i)}, d^{(i)}, M^{(i)}, \sigma^{(i)}) = 1$ *for* $i = 1, 2$,
3. *at most $(d^{(1)} + d^{(2)} - 1)$ public keys in $\mathcal{Y}^{(1)} \cup \mathcal{Y}^{(2)}$ has been queried to $\mathcal{CO}$, and*
4. *Link*$(\sigma_{aux}^{(1)}, \sigma_{aux}^{(2)}) = 0$.

We denote by $\mathbf{Adv}_{\mathcal{A}}^{Link}(\lambda)$ the probability of $\mathcal{A}$ winning the game.

Definition 32 (Linkability.) *An LRS scheme is linkable if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{Link}(\lambda)$ is negligible.*

Non-Slanderability

Definition 33 (Game N-Sland.) *Non-Slanderability for LTRS schemes is defined in the following game between the Simulator $\mathcal{S}$ and the Adversary $\mathcal{A}$ in which $\mathcal{A}$ is given access to oracles $\mathcal{JO}$, $\mathcal{CO}$ and $\mathcal{SO}$:*

1. (Initialization Phase.) *$\mathcal{S}$ generates and gives $\mathcal{A}$ the system's parameters* param.

2. (Probing Phase I.) *$\mathcal{A}$ may query the oracles according to any adaptive strategy.*

3. (Challenge Phase.) *$\mathcal{A}$ gives $\mathcal{S}$ an event id $e^* \in \{0,1\}^*$, a group size $n^* \in \mathbb{N}$ of size polynomial in λ, a threshold $d^* \in [1, n^*]$, a set $\mathcal{Y}^*$ of n^* public keys, a message $M^* \in \{0,1\}^*$ and a set $\mathcal{V}^* \subseteq \mathcal{Y}^*$ of size d^*. No public key in $\mathcal{V}^*$ has been queried to $\mathcal{CO}$ or included in the insider set $\mathcal{V}$ of any query to $\mathcal{SO}$. $\mathcal{S}$ returns a valid augmented signature σ_{aug}^* signed by users with public keys in $\mathcal{V}^*$.*

4. (Probing Phase II.) *$\mathcal{A}$ queries the oracles adaptively, except that no public key in $\mathcal{V}^*$ can be queried to $\mathcal{CO}$ or included in the insider set $\mathcal{V}$ of any query to $\mathcal{SO}$.*

5. (End Game.) *$\mathcal{A}$ gives $\mathcal{S}$ a group size $\hat{n} \in \mathbb{N}$ of size polynomial in λ, a threshold $\hat{d} \in [1, \hat{n}]$, a set $\hat{\mathcal{Y}}$ of public keys that are query outputs of $\mathcal{JO}$ of size $\hat{d}$, a message $\hat{M} \in \{0,1\}^*$ and a signature $\hat{\sigma} \in \Sigma$. Define $\hat{\sigma}_{aux} \doteq (\mathsf{param}, e^*, \hat{n}, \hat{d}, \hat{\mathcal{Y}}, \hat{M}, \hat{\sigma})$.*

In the above game, $\mathcal{A}$ wins if:

1. $\mathsf{Verify}(\mathsf{param}, e^*, \hat{n}, \hat{d}, \hat{\mathcal{Y}}, \hat{M}, \hat{\sigma}) = 1$

2. $\mathsf{Link}(\sigma^*_{aux}, \hat{\sigma}_{aux}) = 1$.

We denote by $\mathbf{Adv}^{NS}_{\mathcal{A}}(\lambda)$ the probability of $\mathcal{A}$ winning the game.

Definition 34 (Non-Slanderability) *An LTRS scheme is* non-slanderable *if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}^{NS}_{\mathcal{A}}$ is negligible.*

Security

Summarizing we have:

Definition 35 (Security of LTRS Schemes.) *An LTRS scheme is* secure *if it is unforgeable, linkably-anonymous, linkable and non-slanderable.*

The model for a Linkable Ring Signature (LRS) is a straightforward special case of the above when the threshold value d is always 1. For schemes that do not support events, one may simply assume they support a single event-id. Therefore, the above model incorporates linkable ring signature schemes no matter they support thresholding and/or events or not.

4.4 Conclusion

In this chapter, we have investigated in depth the linkable ring signatures. We have given an introduction of linkable ring signatures and the significance of linkability in ring signatures. We have introduced notions novel to linkable ring signatures, together with discussion on their importance and impact. Finally we have given a fully developed security model that captures the security requirements of linkable ring signature schemes under various possible adversarial attacks.

Chapter 5

Short Linkable Ring Signatures

In this chapter, we propose the first short linkable ring signature scheme construction. By short we mean the signature size is independent of the size of the member group a signature is signed on behalf of. Being short enables linkable ring signatures to be scalable and deployed in large-scale scenarios. We propose a new mathematical assumption and then reduce the security of our construction to it plus several well-known mathematical assumptions.

5.1 Introduction

Signature size is a crucial factor for group-oriented signature schemes. Usually it is measured as the rate of growth in size with respect to the group size[1]. For example, we say a signature scheme is of signature size $O(n)$ if the signature size grows linearly with the group size n; a scheme is of $O(1)$ if the signature size is a constant and independent of n (in which case we call the scheme "short"). Being short is the key to scalability because the signature size can be kept small even if the group size grows extensively. In other words, a signature scheme that is not short may be practical in a small-group setting, it is unsuitable to be deployed in applications in which a large group is expected.

[1]Of course, there could be other factors affecting the group size. For example, in some construction of threshold group/ring signature schemes, the signature size grows linearly also with the threshold value.

Example 5 Consider the case when a linkable ring signature with linearly-growing signature size is used in electronic voting in order to authenticate a ballot. Assume the signature size is 0.5 KB per group member[2]. If 100 people are involved in a poll within the conference room, then each signature will have an acceptable size of 50 KB. However, 500 MB will be occupied by a single signature if 1 million people are involved, in a community-wide election, for example. Not only is it very demanding for a voter to cast his ballot, it is also a headache for the authority to securely maintain all the 1 million bulky ballots.

Our Contributions in this Chapter

Our contribution in this chapter includes the following:

- We extend the short ring signature scheme construction of Dodis, et al. [43] to the first short linkable ring signature scheme construction.
- We introduce a new hardness assumption, the Link Decisional RSA (LD-RSA) Assumption, and prove the security of our proposed scheme to the assumption as well as some other well-established hardness assumption.

5.2 The Construction

In this section, we give a concrete linkable ring signature (LRS) scheme construction . We then show that such the construction is secure under the security model defined in Chapter 4.

- Init. On input security parameter 1^{λ}, the algorithm prepares a collision-resistant accumulator with one-way domain according to Dodis et al. (refer to section 2.4.6 for details). Define desc to be the description of the

[2]The value is estimated by assuming 4 strings of 1024 bits are needed per group member.

accumulator. The algorithm picks a random generator $\tilde{g} \in QR(N)$ for the group $QR(N)$, where N is defined in desc, and outputs the system's parameters $\mathsf{param} := (1^\lambda, \mathsf{desc}, \tilde{g})$.

- Key-Gen. On input the system's parameters param, the algorithm parses param into $(1^\lambda, \mathsf{desc}, \tilde{g})$ and then executes the probabilistic sampling algorithm W of the accumulator to obtain $(y_i, (p_i, q_i))$. Finally it outputs the key pair (sk_i, pk_i), where $sk_i := (p_i, q_i)$ and $pk_i := y_i$.

- Sign. On input the system's parameters param, a group size $n \in \mathbb{N}$ of size polynomial in λ, a public key set $\mathcal{Y} = \{pk_1, \ldots, pk_n\}$ where all pk_i's are in $\mathcal{PK}$, a message $M \in \{0,1\}^*$ and a private key sk_π that corresponds to $pk_\pi \in \mathcal{Y}$, the algorithm parses parses param into $(1^\lambda, \mathsf{desc}, \tilde{g})$, each pk_i into its respective y_i and sk_π into its respective (p_π, q_π). It then does the following:

  1. Compute the witness w for y_π as $w := f(u, \{y_i | i \in [1,n] \backslash \pi\})$ and then the accumulated value v of all public keys as $v := f(w, y_\pi)$.
  2. Compute a signature for

$$SPK\left\{\begin{pmatrix} w, x, \\ e_1, e_2 \end{pmatrix} : \begin{array}{rcl} w^x & = v \bmod N & \wedge \\ x & \in S(2^\ell, 2^\mu) & \wedge \\ \tilde{y} & = \tilde{g}^{e_1+e_2} & \end{array} \begin{array}{rcl} x & = 2e_1e_2 + 1 & \wedge \\ e_2 & \in S(2^{\ell/2}, 2^\mu) & \wedge \end{array}\right\}(M). \tag{5.1}$$

     The above SPK is instantiated as follows. Randomly pick $r \in_R [0, N/4]$. Compute $a_1 := xr$ and $a_2 := e_2 r$ in $\mathbb{Z}$, and $T_1 := g^r$, $T_2 := g^x h^r$, $T_3 := g^{e_2} s^r$, $T_4 := wy^r$, $T_5 := g^{e_1} t^r$ and $\tilde{y} := \tilde{g}^{e_1+e_2}$ in $QR(N)$. $\tilde{y}$ is the linkability tag. Then execute the following SPK:

  3. Denote by σ' be the output after the execution of the SPK above.

$$SPK\left\{\begin{pmatrix} r, x, \\ e_1, e_2 \end{pmatrix} : \begin{array}{rlcrlc} T_1 & = g^r & \wedge & T_2 & = g^x h^r & \wedge \\ T_1^x & = g^{a_1} & \wedge & T_1^{e_2} & = g^{a_2} & \wedge \\ T_3 & = g^{e_2} s^r & \wedge & T_4^x & = v y^{a_1} & \wedge \\ T_5^{2e_2} g & = g^x t^{2a_2} & \wedge & \tilde{y} & = \tilde{g}^{e_1+e_2} & \wedge \\ x & \in S(2^\ell, 2^\mu) & \wedge & e_2 & \in S(2^{\ell/2}, 2^\mu) & \end{array}\right\}(M). \tag{5.2}$$

The signature σ returned by the algorithm is given by

$$\sigma := (v, T_1, \ldots, T_5, \tilde{y}, \sigma').$$

- Verify. On input the system's parameters param, a group size $n \in \mathbb{N}$ of size polynomial in λ, a public key set $\mathcal{Y} = \{pk_1, \ldots, pk_n\}$ with $pk_i \in \mathcal{PK}$ for all $i \in [1, n]$, a message $M \in \{0,1\}^*$ and a signature $\sigma \in \Sigma$, the algorithm parses param into $(1^\lambda, \mathsf{desc}, \tilde{g})$, each pk_i into its respective y_i, and σ into $(v, T_1, \ldots, T_5, \tilde{y}, \sigma')$. It then verifies the statement $v \stackrel{?}{=} f(u, \{y_i | i \in [1, n]\})$ and the validity of σ' with respect to the SPK represented by Equation (5.2). It returns accept if both checks pass and reject otherwise.

- Link. Given two valid signatures the algorithm extracts their respective linkability tag $\tilde{y}_1$ and $\tilde{y}_2$. It returns linked if they are the same and unlinked otherwise.

Correctness

Verification correctness is straightforward. It is basically implied by the correctness of the SPK. Linking correctness is also easy to see by noting the fact that the linkability tag $\tilde{y}$ is unique for unique (e_1, e_2) in the secret key, with overwhelming probability in λ.

5.3 Security Analysis

5.3.1 Security Theorems

Lemma 1 (Unforgeability.) *Our construction is unforgeable under the Strong RSA assumption in the random oracle model.*

Lemma 2 (Linkable-anonymity.) *Our construction is anonymous under the DDH over $QR(N)$ assumption and LD-RSA assumption in the random oracle model.*

Lemma 3 (Linkability.) *Our construction is linkable under the Strong RSA assumption in the random oracle model.*

Lemma 4 (Non-slanderability.) *Our construction is non-slanderable under the Strong RSA assumption in the random oracle model.*

Summarizing, we have

Theorem 4 (Security.) *Our construction is a secure LRS scheme.*

5.3.2 Proofs

Proof 3 (Lemma 1.) (Sketch.) *The proof follows the same arguments as those in [43]. Simply speaking, if there exists an adversary who is able to produce a valid signature, then he must know a witness behind the SPK represented by Equation (5.2), under the Strong RSA Assumption. Next, the collision-resistant accumulator with one-way domain forces that particular witness to be a secret key whose corresponding public key is in the public key set, due to again the Strong RSA Assumption. In order words, a simulator can be constructed in a way such that if there exists an algorithm that is able to produce a valid signature, the simulator can use it to compute the secret key behind a public key, i.e. solve the strong RSA problem, which contradicts to the strong RSA Assumption.* □

Proof 4 (Lemma 2.) (Sketch.) *Our proposed construction has linkable anonymity because of the following two reasons. First, the SPK represented by Equation (5.2) is a signature scheme derived from an HVZK protocol. Transcripts of HKZK protocols are has zero-knowledge about the witness and thus contain no information about the identity of the actual signer. Second, the auxiliary information* $T_1, \ldots, T_5$ *and* $\tilde{y}$ *are computationally indistinguishable from randomly generated elements if the DDH Assumption and the LD-RSA Assumption hold in* $QR(N)$*. From the above two points, if there exists an algorithm that is enable to distinguish the identity of the actual signer better than random guess, a simulator can be constructed to solve either the DDH Problem in* $QR(N)$ *or the LD-RSA Problem in* $QR(N)$*, contradicting to the Assumptions.* □

Proof 5 (Lemma 3.) (Sketch.) *The proof is somewhat similar to that for unforgeability. Given the Strong RSA Assumption holds, an adversary is forced to compute a signature honestly for the SPK represented by Equation (5.2). As a result, two valid signatures signed using the same secret key are forced to contain the same linkability tag* $\tilde{y}$*. In other words, if there exists an algorithm that is enable to compute two valid but unlinked signature given only one secret key, a simulator can be constructed to solve the Strong RSA Problem.* □

Proof 6 (Lemma 4.) (Sketch.) *If an adversary is able to produce a valid signature that is linked to another signature signed by some honest user, he must know the discrete logarithm* $(e_1 + e_2)$ *of the linkability tag* $\tilde{y}$*. Again, it is due to the soundness of the SPK represented by Equation (5.2), under the strong RSA Assumption. Therefore a simulator can be constructed to solve the Strong RSA Assumption given access to the adversary.* □

Proof 7 (Theorem 4.) *The theorem is a direct consequence of the above lemmas.* □

5.4 Discussion

Devising the Linkability Tag

The most crucial contribution in devising the construction in this chapter is a good design of the linkability tag. Let's look at some criteria that we must take account into when designing the tag.

It should be easy to understand that the tag must be some one-way image of a user's secret key. Intuitively, a linkability tag is like a garbage-like serial number of each user. The one-wayness is to make sure no one can, infer the identity of the actual signer from the tag. However, the above is not enough, as illustrated by the following.

- If we change the mapping from $\tilde{y} := \tilde{g}^{e_1+e_2}$ to $\tilde{y} := \tilde{g}^{e_1}$, then linkability is lost because a single user in possession of a secret key (e_1, e_2) can produce two unlinked signatures with $\tilde{y}_1 := \tilde{g}^{e_1}$ and $\tilde{y}_2 := \tilde{g}^{e_2}$.

- If we replace it with $\tilde{y} := \tilde{g}^{e_1 e_2}$, then L-anonymity is lost because it is easy to identify the actual signer from the tag: $\tilde{g}^{y_i} \stackrel{?}{=} \tilde{y}^2\tilde{g}$.

However, a (probably) secure alternative choice is $\theta_{d,4}((e_1, e_2)) = (g_\theta^{e_1}, g_\theta^{e_2})$. Note that $\theta_{d,2}$ while $\theta_{d,3}$ are not special PK-bijective.

A related security requirement is to that, given a random sample y_1, it is hard to compute y_2 such that there exist x_1, x_2, satisfying (x_1, y_1), (x_2, y_2) $\in \mathcal{R}$, $\theta_d(x_1) = \theta_d(x_2)$. This stronger concept may be needed in further study of the current topic, but it is not needed in the present paper.

Short Linkable Group Signature

It is straightforward to extend our short linkable ring signature construction to *linkable group signatures* [76, 77]. Simply also escrow the user identity (or the user public key) to an Open Authority (OA) in the signatures. The escrow

can be done by verifiably encrypt the identity (or public key) to the OA by methods in [9], for example.

5.5 Conclusion

In this chapter, we have proposed the first short linkable ring signature scheme construction. By short we mean the signature size is independent of the size of the member group a signature is signed on behalf of. Being short enables linkable ring signatures to be scalable and deployed in large-scale scenarios. We have proposed a new mathematical assumption and then reduced the security of our construction to it plus several well-known mathematical assumptions.

Chapter 6

Separable Linkable Threshold Ring Signatures

In this chapter, we propose the first separable linkable threshold ring signature scheme. Separability is the key for a scheme to be practically deployed in ad hoc environment in which machines are highly heterogenous. Our proposed scheme also supports thresholding efficiently in the sense of computational and storage/communication complexities. We reduce the security of our scheme to well-known mathematical assumptions.

6.1 Introduction

THRESHOLD CRYPTOGRAPHY. As have discussed before, threshold cryptography aims at extending conventional cryptographic protocols into a multi-user setting. In the sense of (linkable) ring signature schemes, it means the following: among a group of n members, some $t \in [1, n]$ of them[1] work jointly to sign a signature. Such a signature is verified to be valid if and only if t or more members cooperate in the signing process. Thus a verifier is convinced by a valid signature that some t or more members out of a group of n users agreed to sign.

[1]The special case when $t = 1$ actually goes back to the conventional non-threshold case.

In [68], a (d, n)-threshold extension to its original linkable ring signature scheme is constructed by concatenating d linkable ring signatures. We note that the construction, though simple and trivial, is not efficient. In particular, the space and time complexities are both $O(dn)$. We give in this chapter a construction with time and space complexities both being $O(n)$.

SEPARABILITY. In [23], Camenisch, et. al. diversified the concept of separability of cryptographic protocols [63] into *perfect separability*, *strong separability* and *weak separability* when describing users' ability to choose their own cryptographic primitive and system parameters. Separability is of particular importance for ring signature schemes as there is no group manager to coordinate the choice of signature primitive and system parameters for each user. For instance, a ring signature scheme that is only weakly separable is not practical at all as it is unlikely to have all group members using the same primitive, system parameters and security parameters. The RSA implementation of [91, 1, 67, 100, 68] are strongly separable while the DL implementation of [1, 67, 68] are only weakly separable.

Our Contributions in this Chapter

Our contribution in this chapter includes:

- We give a construction of the first separable linkable ring signature scheme. It also the first linkable ring signature scheme construction of the CDS-type ([37]).
- Our construction supports bandwidth-efficient threshold signing. The signature size in [68] is $O(dn)$ while ours is $O(n)$, where n is the number of users and d is the threshold.
- We prove the security of our construction based on well-known hard problem assumptions.

6.2 The Construction

In this section, we give a concrete construction of a linkable threshold ring signature (LTRS) scheme. We then show that such a construction is secure under the security model defined earlier.

- Init. On input the security parameter $1^\lambda \in \mathbb{N}$, the algorithm picks $\kappa \in \mathbb{N}$ of size polynomial in λ and $1 < \epsilon \in \mathbb{R}$. It also picks a κ-bit prime q uniformly at random and then a strong collision-resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_q$. It outputs the system parameters $\mathsf{param} = (1^\lambda, \epsilon, \kappa, q, H)$.

- Key-Gen. On input the system's parameters param and a further security parameter 1^{λ_i} such that $\lambda_i \geq \lambda$, the algorithm randomly picks two distinct primes p_i, q_i of the form $p_i = 2p_i' + 1$ and $q_i = 2q_i' + 1$, where p_i', q_i' are both $((\lambda_i - 2)/2)$-bit primes, and sets $N_i := p_i q_i$. It then picks uniformly at random an element $g_i \in_R \{g | \langle g \rangle = QR(N_i)\}$ and an integer $x_i \in_R \mathbb{Z}_{p_i' q_i'}$ and computes $y_i := g_i^{x_i}$. It picks a strong collision-resistant hash function $H_i : \{0,1\}^* \to \{h | \langle h \rangle = QR(N_i)\}$. It sets the public key to $pk_i := (1^{\lambda_i}, N_i, g_i, y_i, H_i)$, and the secret key to $sk_i := (p_i, q_i, x_i)$. Finally it outputs (sk_i, pk_i).

- Sign. On input the system's parameters $\mathsf{param} = (1^\lambda, \epsilon, \kappa, q, H)$, an event-id $e \in \{0,1\}^*$, a group size $n \in \mathbb{N}$ of size polynomial in λ, a threshold $d \in [1, n]$, a public key set $\mathcal{Y} = \{pk_1, \ldots, pk_n\}$ where each $pk_i = (1^{\lambda_i}, N_i, g_i, y_i, H_i)$ is s.t. $\lambda_i \geq \lambda$, a message $M \in \{0,1\}^*$, and a private key set $\mathcal{X} = \{sk_{\pi_1}, \ldots, sk_{\pi_d}\}$ where each $sk_{\pi_i} = (p_{\pi_i}, q_{\pi_i}, x_{\pi_i})$ corresponds to $pk_{\pi_i} \in \mathcal{Y}$, the algorithm does the following: (Define $\mathcal{I} \doteq \{\pi_1, \ldots, \pi_d\} \subseteq [1, n]$.)

  1. For all $i \in [1, n]$, compute the *tag bases* $h_{i,e} := H_i(\mathsf{param}, pk_i, e)$ and the *tags*

$$\tilde{y}_{i,e} := \begin{cases} h_{i,e}^{x_i}, & i \in \mathcal{I}; \\ h_{i,e}^{a_i}, & i \in [1,n] \backslash \mathcal{I},\ a_i \in_R \mathbb{Z}_{\lfloor N_i/4 \rfloor}. \end{cases}$$

2. Compute a signature $(f, s_1, \ldots, s_n)$ for

$$SPK\left\{(\alpha_1, \ldots, \alpha_n) : \bigvee_{\mathcal{J} \in \wp_d([1,n])} \left(\bigwedge_{i \in \mathcal{J}} y_i = g_i^{\alpha_i} \wedge \tilde{y}_{i,e} = h_{i,e}^{\alpha_i} \right) \right\}(M).$$

In particular, this requires the knowledge of $x_{\pi_1}, \ldots, x_{\pi_d}$. We will refer to this signature scheme as SPK_1.

3. Compute a signature $(c, s'_1, \ldots, s'_n)$ for

$$SPK\left\{(\beta_1, \ldots, \beta_n) : \bigwedge_{i=1}^{n} \tilde{y}_{i,e} = h_{i,e}^{\beta_i} \right\}(M).$$

In particular, this requires the knowledge of x_i for all $i \in \mathcal{I}$ and a_i for all $i \in [1,n] \backslash \mathcal{I}$. We will refer to this signature scheme as SPK_2.

4. Output the signature as

$$\sigma := \langle ((\tilde{y}_{1,e}, \ldots, \tilde{y}_{n,e}), (f, s_1, \ldots, s_n), (c, s'_1, \ldots, s'_n) \rangle.$$

Remark: The signature is composed of three parts: (1) a set of tags, (2) a signature for SPK_1, and (3) a signature for SPK_2.

- Verify. On input the system's parameters $\mathsf{param} = (1^\lambda, \epsilon, \kappa, q, H)$, an event id $e \in \{0,1\}^*$, a group size n of size polynomial in λ, a threshold $d \in [1,n]$, a public key set $\mathcal{Y} = \{pk_1, \ldots, pk_n\}$ where each $pk_i = \lambda_i, N_i, g_i, y_i, H_i)$ with $\lambda_i \geq \lambda$, a message $M \in \{0,1\}^*$, and a signature $\sigma \in \Sigma$, the algorithm parses σ into $\langle ((\tilde{y}_1, \ldots, \tilde{y}_n), (f, s_1, \ldots, s_n), (c, s'_1, \ldots, s'_n) \rangle$.

1. For $i \in [1,n]$, compute $h_{i,e} := H_i(\mathsf{param}, pk_i, e)$.

2. Verify if $(f, s_1, \ldots, s_n)$ is a correct signature for SPK_1.
3. Verify if $(c, s'_i, \ldots, s'_n)$ is a correct signature for SPK_2.
4. Output 1 if the above verifications are okay. Output 0 otherwise.

- Link. On input two augmented signatures $\sigma_{aux}^{(1)}$ and $\sigma_{aux}^{(2)}$ such that $\mathsf{Verify}(\sigma_{aux}^{(i)}) = 1$ for $i = 1, 2$, the algorithm parses, for $i = 1, 2$, $\sigma_{aux}^{(i)}$ into

$$(\mathsf{param}^{(i)}, e^{(i)}, n^{(i)}, d^{(i)}, \mathcal{Y}^{(i)}, M^{(i)}, \sigma^{(i)}).$$

It is required that $\mathsf{param}^{(1)} = \mathsf{param}^{(2)}$ and $e^{(1)} = e^{(2)}$. The algorithm parses, for $i = 1, 2$, $\mathcal{Y}^{(i)}$ into $\{pk_1^{(i)}, \ldots, pk_{n^{(i)}}^{(i)}\}$ and $\sigma^{(i)}$ into

$$\langle((\tilde{y}_{1,e}^{(i)}, \ldots, \tilde{y}_{n^{(i)},e}^{(i)}), (f^{(i)}, s_1^{(i)}, \ldots, s_{n^{(i)}}^{(i)}), (c^{(i)}, s_1'^{(i)}, \ldots, s_{n^{(i)}}'^{(i)})\rangle.$$

If there exists $\pi^{(1)} \in [1, n^{(1)}]$ and $\pi^{(2)} \in [1, n^{(2)}]$ s.t. $pk_{\pi^{(1)}}^{(1)} = pk_{\pi^{(2)}}^{(2)}$ and $\tilde{y}_{\pi^{(1)},e}^{(1)} = \tilde{y}_{\pi^{(2)},e}^{(2)}$, it returns 1 and additionally $pk_{\pi^{(1)}}^{(1)}$. Otherwise it returns 0.

Correctness

Straightforward.

6.3 Security Analysis

In this section, we prove the security of our construction.

6.3.1 Security Theorems

We have the following lemmas.

Lemma 5 (Unforgeability.) *Our construction is unforgeable under the Strong RSA assumption in the random oracle model.*

Lemma 6 (Linkable-anonymity.) *Our construction is anonymous under the Strong RSA assumption and DDH over QR(N) assumption in the random oracle model.*

Lemma 7 (Linkability.) *Our construction is linkable under the Strong RSA assumption in the random oracle model.*

Lemma 8 (Non-Slanderability.) *Our construction is non-slanderable under the Strong RSA assumption in the random oracle model.*

Finally, we have the following theorem.

Theorem 5 (Security.) *Our construction is a secure LTRS scheme.*

6.3.2 Proofs

Proof 8 (Lemma 5.) *(Sketch.) Roughly speaking, similarly constructed ring signatures [67] already has unforgeability, and that implies unforgeability with linkable ring signatures.* □

Proof 9 (Lemma 6.) *(Sketch.) Simulating Signing Oracle, $\mathcal{SO}$: Upon input $(e, n, d, \mathcal{Y}, \mathcal{V}, \mathcal{X}, M)$, generate a valid signature as follows: For each $i \in \mathcal{Y} \backslash \mathcal{V}$, randomly generate a_i and compute $\tilde{y}_{i,e} = h_{i,e}^{a_i}$. For each $i \in \mathcal{V}$, randomly generate a_i and backpatch the random oracle to $h_{i,e} = H_i(\mathsf{param}, pk_i, e) = g_i^{a_i}$ and compute $\tilde{y}_{i,e} = y^{a_i}$. Ensure consistency with other oracles from the beginning. Generate $c_0, \cdots, c_n$ such that they interpolate a polynomial f with degree $\leq n - d$ and $f(i) = c_i$ for $0 \leq i \leq n$. For each i, simulate the corresponding 3-move conversation in Step (2) of* Sign *with randomly generated responses $s_1, \cdots, s_n$ to produce the commitments. Backpatch the random oracle so that the commitments are hashed to c_0. This completes up to Step (2) in* Sign*. The rest is easy: Randomly generate challenge c, simulate the SPK in Step (3) of* Sign *with randomly generate responses $s'_1, \cdots, s'_n$.*

Setting up the gauntlet for solving DDH: Similar to proof of anonymity in [68]. Let Q_J be the number of $\mathcal{JO}$ queries. Denote the Gauntlet DDH Problem as $(\hat{N}, \hat{g}, \hat{g}^{\alpha}, \hat{g}^{\beta}, \hat{g}^{\gamma})$ where $\gamma = \alpha\beta$ with probability 1/2. In the Gauntlet Phase, Simulator $\mathcal{S}$ sets up the witness extraction mechanism as follows: Randomly select $i^* \in \{1, \cdots, Q_J\}$. Return $pk^* \leftarrow (\hat{l}, \hat{N}, \hat{g}, \hat{g}^{\alpha}, \hat{H})$ in the i^*-th $\mathcal{JO}$ query, backpatch Random Oracle $\mathcal{HO}_{i^*}$ to $h_{i,e} = \hat{g}^{\beta}$. There is a non-negligible probability that $pk^* = y_g$, the gauntlet public key. Generate the Gauntlet signature σ'_g with $\tilde{y}_{i,e} = \hat{g}^{\gamma}$ and simulate the SPK's. With 1/2 probability, $\alpha\beta = \gamma$ and it can be shown that the gauntlet signature is indistinguishable from one generated using Sign*. Otherwise, with 1/2 probability, $\alpha\beta \neq \gamma$ and it can be shown that σ'_g is indistinguishable from one generated using $\mathcal{SO}$.*

If $\mathcal{A}$ returns $\hat{b} = 1$, $\mathcal{S}$ answers Yes *to the DDH question. Otherwise, $\mathcal{S}$ answers* No*. $\mathcal{S}$'s advantage in DDH equals $\mathcal{A}$'s advantage in winning Game LA.* □

Proof 10 (Lemma 7.) *(Sketch.) Similar to proof of linkability in [68]. If Adversary can produce two unlinked signatures, then he is rewound twice to produce two sets of witnesses of set-size d_1 and d_2 respectively. If the two sets overlap, then the threshold signatures should have already been linked. If the two sets do not overlap, then we would have obtained a total of $d_1 + d_2$ witnesses while Adversary only corrupted at most $d_1 + d_2 - 1$ witnesses.* □

Proof 11 (Lemma 8.) *(Sketch.) The non-slanderability is protected by Step (3) of the signature. Given a signature from $\mathcal{SO}$, Adversary does not know the discrete logarithm of any $\tilde{y}_i$, and therefore cannot produce a signature containing some $\tilde{y}_j$ and prove knowledge of logarithm of $\tilde{y}_j$ as in* Sign*'s Step (3).* □

Proof 12 (Theorem 5.) *The proof is a straightforward implication of the above lemmas.* □

6.4 Discussion

Separability

In our LRS scheme construction, individual users can choose their own security parameter λ_i (as long as it is no less than the global security parameter λ), their own λ_i-bit safe-prime product N_i and also their own group generator g_i for $QR(N_i)$. As a result, our construction is separable.

We used in our construction user key pairs from the Discrete Logarithm (DL) over composite moduli, i.e. the secret key x_i and public key y_i of user i are related by $y_i = g_i^{x_i} \pmod{N_i}$, where N_i is a composite modulus. However it is possible and straightforward to modify our construction to allow user key pairs from DL over a prime modulus, i.e. the keys are related by $y_i = g_i^{x_i} \pmod{P_i}$, P_i being prime. Of course, the security will then reduce to different hardness assumptions, namely the DL and DDH Assumptions over finite cyclic groups. Putting it one step further, our construction can actually support a mixture of composite DL and prime DL key pairs.

RST-type Ring Signatures

Our construction utilizes the CDS-type structure, meaning the structure from Cramer, et al. [37]. However it is easy to adapt the technique in our construction to construct the first separable linkable ring signature of the RST-type, meaning the structure from Rivest, et al. [91], if thresholding is not required.

The idea of how to do it is to simply follow the construction given by [68], but use different tags for different users instead using a single tag for all users. If we denote by $\tilde{y}_i$ the tag for user i, then $\tilde{y}_i := h_i^{a_i}$ for some randomly generated a_i except $\tilde{y}_s := h_s^{x_s}$ with signer s. All the signer has to do is to simulate the following signature of knowledge

$$SPK\{(x_i) : y_i = g_i^{x_i} \wedge \tilde{y}_i = h_i^{x_i}\}(M)$$

along the *ring*, but with the challenge computed as $Hash(commitments_i) := challenge_{i+1}$, except for the actual signer.

The linkable ring signature scheme construction resulted is still separable, and still supports a mixture of composite DL and prime DL key pairs.

Thresholding

The time and storage/communication complexities of our linkable threshold ring signatures is $O(n)$ (n being the group size), and is thus independent of the threshold value d. This greatly improves upon the construction given by [68] in which the time and storage/communication complexities are $O(dn)$[2].

In [68], a linkable ring signature scheme construction is first presented and its threshold extension is later discussed. There is no security model and proofs for the threshold extension. Even though the extension is surprisingly simple and plausible, there is no in-depth analysis of the security of such an extension. On the contrary, we present our linkable threshold ring signature scheme with full security model and proofs.

However, our scheme is interactive while [68] is non-interactive. More specifically, the signers in our scheme must interact during the signing process. In [68], the signers produce on their own their "partial signatures", which can later be combined by anybody to become a linkable threshold ring signature.

6.5 Conclusion

In this chapter, we have proposed the first separable linkable threshold ring signature scheme. Separability is the key for a scheme to be practically deployed in ad hoc environment in which machines are highly heterogenous. Our

[2]For example, if one takes d to be $n/2$, then their complexities are $O(n^2)$, inferring impracticality for fairly large n.

proposed scheme also supports thresholding efficiently in the sense of computational and storage/communication complexities. We have reduced the security of our scheme to well-known mathematical assumptions.

Chapter 7

Applications

In this chapter, we discuss real-life examples when Cryptography is applied to achieve the stringent and sometimes contradictory requirements of various applications. The three applications we are going to look at are: *Offline Anonymous Electronic Cash*, *E-Voting* and *Anonymous Attestation*.

Our Contributions in this Chapter

Our contributions in this chapter are as follows.

- Kiayias et al. presented in [62] the first electronic voting scheme that simultaneously achieved efficient tallying, universal verifiability and write-in capability, for typical voter distribution under which only a small portion writes in. We discuss that e-voting scheme constructed from linkable ring signatures [68] also achieve the same three properties, even for all worst-case voter distributions.
- We discuss an efficient implementation of anonymous attestation [17] using linkable ring signatures, and the construction of an offline anonymous electronic cash system using linkable ring/group signatures.

7.1 Offline Anonymous Electronic Cash

7.1.1 Introduction

Offline Anonymous Electronic Cash (e-cash) can be thought of the electronic counterpart of conventional paper money and coins. It is the key to the success of e-business because it enables business transactions to be done over the Internet. Some people even believe that e-cash will eventually replace all paper money and coins. There have been a lot of strong incentives to motivate the development of e-cash system – enabling e-business is with no doubt a prime example; others include reducing the cost of printing paper money and stamping coins, offering better protection against fraud and black-mailing, etc.

Despite decades of effort, online business is still far away from being popular. The reasons are complicated. On one hand, devising an efficient and yet secure e-cash system has been proven to be a very difficult task. Not being 100% secure is simply far too risky and thus unacceptable in business and banking industry. On the other hand, end users' and corporation's habit of paying, inertia of switching and skeptic attitude towards new technology greatly undermine the possible development of new systems.

A major stream of e-cash systems found in the literature makes use of *blind signatures*. In such systems, the users withdraw electronic coins that consist of numbers generated by users and then blindly signed by the bank. Each blind signature then represents a given amount. When these coins are later spent in shops, the merchants can authenticate the coins by using the public key of the bank. Anonymity of users is maintained in the transactions as nobody, not even the bank, can link the withdrawed coins and the spent coins. Existing schemes of this category are fruitful, some of the important ones are: [30, 33, 15, 24].

E-cash systems by *group signatures* recently received much attention. The group members in the group signature scheme forms a group of users. The

bank, who plays the role of the Group Manager, is capable of issuing electronic coins (which are actually the member certificates, or the ability to sign) to the users. When a user spends, what he/she does is to sign a group signature for the shop. The anonymity inherited from the group signature scheme provides privacy for the users. Examples: [72, 96, 73].

7.1.2 Construction

The short linkable ring signature scheme construction we proposed in Chapter 5 can be used to construct an e-cash scheme. It serves as a new alternative to e-cash schemes of the "group signature approach", as described in the introduction.

The Bank takes the role of the Group Manager. We adopt the "group of coins" model: each user key pair represents a coin; the knowledge of a user secret key means the ability to spend a coin; and anonymity is among the group of coins issued. The Bank initializes our short linkable group signature scheme. Assume the shops and the users have their accounts established with the bank.

- *(Withdrawal.)* To withdraw a coin, the user first runs the Key Generation algorithm to obtain a key pair. He keeps the secret key with himself and gives the public key to the bank. The bank debits the user's account, and update the group public key by accumulating the new public key into the current group public key.

- *(Payment.)* The user signs a linkable ring signature, using his secret key, on the payment transcript, on behalf of the most up-to-date coin group (i.e. using the most up-to-date group public key). The shop verifies against the signature and accepts the payment if the signature is valid.

- *(Deposit.)* The shop gives the bank the payment transcript, along with

the associated linkable group signature. The bank verifies as the shop did and credits the shop's account if the signature is valid. To detect double-spending, the bank goes through the deposit database to look for signatures that are linked.

Double spenders of the e-cash are detected as double signers of the linkable ring signature scheme. However, methodologies differ after detection. In *non-accusatory* linkability, the suspect can only be *tagged* and prevented from further double spending afterwards. The drawbacks are time delay to effective tagging and small punishment for the offense. In *accusatory* linkability, the linking algorithm outputs a suspect. If linkable group signature is used instead, the linked signatures can be passed on to the revocation manager in order to open the identity of the double spender.

7.2 Electronic Voting

7.2.1 Introduction

Electronic Voting is to vote with the help of computers. Situations can vary from replacing paper-ballots with touch-screen terminals, to enabling a voter to cast his vote at home. Electronic Voting tries to make voting easier and more accessible, tallying faster and more accurate, the overall cost lower, the poll more secure, etc.

From the technology point of view, cryptographically secure ballot elections can be classified into three basic paradigms. Under the *blind signature* [30] paradigm, the voters get their ballots from the voting authority, in a certified but privacy-preserved way. This enables them to embed any form of ballot, e.g. multiple choice questions, open-ended questions, or both. An anonymous channel is required between the voter and the tallying authorities in order to hide the identity of the voter when he casts his ballot. In this

approach, universal verifiability is missing and robustness is achieved usually by thresholding the authority.

Under the *homomorphic encryption* [36] paradigm, the casted ballots are first encrypted and then "compressed" using a homomorphic encryption scheme into a tally. This compression property allows extremely fast tallying, and is one of the reasons why this approach is attractive. The drawback is, however, homomorphic encryption can only compress ballots without write-in and is therefore only suitable when ballots contain only multiple choice questions.

Under the *mix-net* [29] paradigm, the tallying officials move the ballots between them and permute them in the process while changing their representation (e.g., partially decrypting them). Practical implementations of this approach in its fully robust form is still considered a slow tallying process.

Remarkable advances in group/ring signatures in recent years have given new options to e-voting scheme constructions. In fact, many papers on group/ring signatures have included e-voting as applications. Using group/ring signatures contributes to a new paradigm of e-voting construction.

Nevertheless, none of the existing group/ring signature schemes gives raise to a satisfactory construction. First, most group signature schemes are unlinkable, which means double-voting cannot be detected (an exception: the one-show credential system due to [19]). Secondly, and more importantly, anonymity revocation is an inherited property in group signatures/credential systems. Note that anonymity is of prime concern in e-voting. Nothing justifies to open a vote.

Previously proposed linkable ring signature schemes partly solved the problem because they have (1) double-voting detecting capability and (2) no anonymity revocation. However, all existing schemes have signature sizes linear with the signing group, which makes them impractical when used in large-scale voting. Our short linkable ring signature scheme construction given in Chapter 5 has constant signature size and is thus very practical in this sense.

7.2.2 Construction

We use the construction of an e-voting scheme from [68]. The main contribution of the construction that appears here with respect to the one in [68] is that we have an $O(1)$-sized signature whereas [68] used an $O(n)$-sized signature, where n is the group size. We summarize the e-voting scheme below. For further details, see [68].

- *(Registration.)* Through a registration process, a list of the public keys of all eligible voters is published. Each voter can check if his public key is included. A number of independent registrants can be used to ensure that no ineligible entity is listed.

- *(Vote Casting.)* Each voter sends in a linkable ring signature on a message which states its selected candidate, from a prescribed candidate list or as a write-in candidate. The cast ballots can be listed in a public bulletin board for voter inspection.

- *(Tallying.)* Simply verify all received linkable ring signatures, drop the invalid or linked ones, and tally the remaining according to their signed messages.

Kiayias and Yung [62] hybridized homomorphic encryption and mix-net to achieves simultaneously (1) efficient tallying, (2) universal verifiability and (3) write-in capability under typical voter distribution where only a small proportion of voters write-in. Our e-voting scheme above achieves the same even under worst-case voter distributions: the proportion of voters who write in can vary from negligible to overwhelming. To write-in in our scheme, a voter simply sends in a linkable ring signature on the message which includes its write-in candidate.

If one worries about the group manager having too much power from knowing the factoring of N, then Boneh and Franklin's [12] for generating N collaboratively among a number of servers, none of which knows the factoring of N, can be used. Nakanishi, et al. [76, 77] presented e-voting from linkable group signature. Our version of the linkable group signature can also be used to construct e-voting.

7.2.3 Discussions

In this section, we would like to talk about electronic voting in a perspective beyond the technology aspect. Although advanced technology is the core enabler of wide-scale electronic voting, there are issues that have nothing (or little) to do with it, but should be taken account into when electronic voting is brought in practice.

The first legally binding online election in the US took place during March of 2000. The Democratic presidential primary in Arizona included the possibility to cast votes using home PCs or sites set up exclusively for this purpose. In presidential election of 2004, some states had new electronic voting systems in operation. Many security analysts warned that computer voting terminals had a significant possibility of voter fraud or data corruption by a software attack. Others said that recounts would be nearly impossible with the machines and criticized the lack of a "paper trail", which is included in many other trivial events such as grocery shopping or using an ATM.

Making sure that every vote counts is undoubtedly a bedrock of democracy. To assure people that their vote is counted as they cast it, the counting of votes has to be as transparent as possible. Obviously paperless electronic voting on touch-screen terminals offers no confidence to voters that votes are counted as they casted. When the software on which votes are counted is protected as a corporate trade secret and the software is so complex that if maliciously or

unintentionally bad code was embedded no analysis could discover it. Further, because there is no voter verified paper record, it is just impossible to do any auditing or recount on the electronic votes. Finally the opportunities for fraud exist on a greater scale than ever before.

7.3 Anonymous Attestation

7.3.1 Introduction

Trusted Computing Group (TCG) develops and promotes open industry standard specifications for trusted computing hardware building blocks and software interfaces across multiple platforms, e.g. PC's, PDA's, and digital phones. This enables more secure data storage, online commerce transactions, etc, while protecting privacy and individual rights. In the context of the (TCG), Anonymous Attestation is a solution to the following problem: The user of such a platform communicates with a verifier who wants to be assured that the user indeed uses a platform containing such a trusted hardware module, i.e., the verifier wants the trusted platform module (TPM) to authenticate itself. However, the user wants her privacy protected and therefore requires that the verifier only learns that she uses a TPM but not which particular one.

The first solution [56] has the drawback of requiring a TTP to be online in every transaction. Also, anonymity is lost when the TTP and the verifier collude. [17] solves the problem by making use of a group signature scheme variant based on the Camenisch-Lysyanskaya group signature scheme [19, 21]. Among other differences from the original scheme, the two crucial ones are (1) disabling anonymity revocation and (2) including a pseudonym in the signatures.

In essence, DAA [17] is a group signature without revocability, and with an additional feature of *rogue tagging*. Double signers can be detected, or linked,

yet their identities are not revealed. When a double signer is detected, a *rogue tag* is produced to prevent it from signing again: future signatures (attestations) identified with a known rogue tag is not accepted. Double signers of different transactions with the same *basename*, bsn, are detected. But signing twice with different basename is not detected.

The linkable ring signature is ideally suited to implementing DAA. It is a group signature without revocation. Its linkability feature can be used to detect double signers, and when linked output the linkability tag, $\tilde{y} = g_\theta^{sk}$, as the rogue tag. Future signatures whose $\tilde{y}$ equals a known rogue tag is not accepted. The value g_θ can be made a function of the basename but not the transaction, e.g. $g_\theta = Hash(\mathsf{bsn}, \cdots)$. Then double signing on different transactions with same basename is linked, while double signing on different basename will not be linked.

7.3.2 Construction

Below, we highlight a few important points in implementing Anonymous Attestation from linkable ring signatures. Readers may refer to [17] for further details.

- *(Setup for Issuer.)* The issuer acts as the Group Manager. He initializes our short linkable ring signature scheme.
- *(Join Protocol.)* The TPM joins by first running Key Generation algorithm of the linkable ring signature scheme in order to obtain a user key pair. It then submits the public key to the Issuer and retains the secret key. It also proves to the Issuer that the public key is correctly formed.
- *(DAA-Signing Protocol.)* The TPM signs a linkable ring signature by invoking the Signing algorithm of the linkable ring signature scheme.

- *(Verification Algorithm.)* This is exactly the same as the Verification algorithm of the linkable ring signature scheme.
- *(Rogue Tagging.)* When a user secret key is found, it should be distributed to all potential verifiers. These verifiers can then put the key on their list of rogue keys.

7.4 Conclusion

In this chapter, we have discussed real-life examples when Cryptography is applied to achieve the stringent and sometimes contradictory requirements of various applications. The three applications we are going to look at are: *Offline Anonymous Electronic Cash*, *E-Voting* and *Anonymous Attestation.*

Chapter 8

Conclusion

In this thesis, we have proposed two linkable ring signature schemes for privacy-preserving applications. They are *short linkable ring signature scheme* and *separable linkable threshold ring signature scheme.* The *short linkable ring signature scheme* is the first linkable ring signature scheme that produces signatures of size independent of group size. This makes the scheme scalable and very practical to be deployed in a large scale. The *separable linkable threshold ring signature scheme* is the first of its kind to support separability and efficient thresholding. Separability allows users of a scheme to be heterogenous from security parameters to cryptographic primitives and therefore is a favorable property in ad hoc networks.

We have discussed and rigorously define notions of security and functionality that have never been considered in the literature, such as *accusatory linking* and *non-slanderability.* Accusatory linking identifies a cheating signer and hence discourages cheating. Accusatorily linkable ring signatures therefore find new applications. Non-slanderability ensures honest users cannot be slandered on. It is a vital property that should be possessed by all linkable ring signature schemes. We have formulated a security model for linkable (threshold) ring signature schemes and prove the security of our two proposed constructions under the model, under some well-known mathematical assumptions and the *Link Decisional RSA* (LD-RSA) Assumption we formulate.

We have investigated three challenging privacy-preserving applications. They are *offline anonymous electronic cash*, *electronic voting* and *anonymous attestation*. They all face a thorny and contradicting difficulty – on one hand users want their privacy to be maintained, on the other the authority wants authentication for eligibility. We have shown how to use our proposed schemes to implement all the three of them.

Appendix A

Paper Derivation

The following is the list of papers derived from this thesis:

1. Patrick P. Tsang, Victor K. Wei, Tony K. Chan, Man Ho Au, Joseph K. Liu, and Duncan S. Wong. *Separable Linkable Threshold Ring Signatures.* In INDOCRYPT 2004, Lecture Notes in Computer Science 3348, pp. 384-398. Springer-Verlag, 2004. (Acceptance rate: 17%)

2. Patrick P. Tsang and Victor K. Wei. *Short Linkable Ring Signatures for E-Voting, E-Cash and Attestation.* In ISPEC 2005, Lecture Notes in Computer Science 3439, pp. 48-60. Springer-Verlag, 2005. (Acceptance rate: 33%)

Bibliography

[1] Masayuki Abe, Miyako Ohkubo, and Koutarou Suzuki. 1-out-of-n signatures from a variety of keys. In *ASIACRYPT 2002*, pages 415–432, 2002.

[2] C. Adams and S.Farrell. Internet x.509 public key infrastructure certificate management protocols. Internet Engineering Task Force: RFC 2510, 1999.

[3] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO 2000*, volume 1880 of *LNCS*, pages 255–270. Springer-Verlag, 2000.

[4] Niko Bârić and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 480–494, 1997.

[5] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In *CRYPTO 1992*, volume 740 of *LNCS*, pages 390–420, 1992.

[6] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 614–629. Springer-Verlag, 2003.

[7] Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM Press, 1993.

[8] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. Cryptology ePrint Archive, Report 2004/077, 2004. `http://eprint.iacr.org/`.

[9] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA 2005*, volume 3376 of *LNCS*, pages 136–153. Springer-Verlag, 2005.

[10] Josh Cohen Benaloh and Michael de Mare. One-way accumulators: A decentralized alternative to digital signatures (extended abstract). In *EUROCRYPT 1993*, volume 765 of *LNCS*, pages 274–285, 1993.

[11] A. Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In *PKC 2003*, volume 2567 of *LNCS*, pages 31–46. Springer-Verlag, 2003.

[12] D. Boneh and M. Franklin. Efficient generation of shared RSA keys. In *CRYPTO 1997*, volume 1294 of *LNCS*, pages 425–439. Springer-Verlag, 1997.

[13] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer-Verlag, 2004.

[14] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 416–432. Springer-Verlag, 2003.

[15] Stefan Brands. Untraceable off-line cash in wallets with observers (extended abstract). In *CRYPTO 1993*, volume 773 of *LNCS*, pages 302–318. Springer-Verlag, 1993.

[16] Emmanuel Bresson, Jacques Stern, and Michael Szydlo. Threshold ring signatures and applications to ad-hoc groups. In *CRYPTO 2002*, volume 2442 of *LNCS*, pages 465–480. Springer-Verlag, 2002.

[17] Ernie Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*, pages 132–145, New York, NY, USA, 2004. ACM Press.

[18] J. Camenisch and M. Michels. A group signature scheme based on an RSA-variant. rs RS-98-27, BRICS, 1998.

[19] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer-Verlag, 2001.

[20] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *CRYPTO 2002*, volume 2442 of *LNCS*, pages 61–76, 2002.

[21] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In *SCN 2002*, volume 2576 of *LNCS*, pages 268–289. Springer-Verlag, 2002.

[22] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO 2004*, volume 3152, pages 56–72, 2004.

[23] Jan Camenisch and Markus Michels. Separability and efficiency for generic group signature schemes. In *CRYPTO 1999*, volume 1666 of *LNCS*, pages 413–430. Springer-Verlag, 1999.

[24] Jan Camenisch, Jean-Marc Piveteau, and Markus Stadler. An efficient fair payment system. In *Proceedings of the 3rd ACM conference on Computer and communications security*, pages 88–94. ACM Press, 1996.

[25] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups (extended abstract). In *CRYPTO 1997*, volume 1294 of *LNCS*, pages 410–424. Springer-Verlag, 1997.

[26] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *STOC*, pages 209–218, 1998.

[27] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *CoRR*, cs.CR/0010019, 2000.

[28] Tony K. Chan, Karyin Fung, Joseph K. Liu, and Victor K. Wei. Blind spontaneous anonymous group signatures for ad hoc groups. In *ESAS 2004*, volume 3313 of *LNCS*, pages 82–94. Springer-Verlag, 2004.

[29] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM 1981*, 24(2):84–90, 1981.

[30] D. Chaum. Blind signatures for untraceable payments. In *CRYPTO 1982*, pages 199–203. Plenum Press, 1982.

[31] David Chaum. Designated confirmer signatures. In *EUROCRYPT 1994*, volume 950 of *LNCS*, pages 86–91. Springer-Verlag, 1994.

[32] David Chaum, Jan-Hendrik Evertse, and Jeroen van de Graaf. An improved protocol for demonstrating possession of discrete logarithms and

some generalizations. In *EUROCRYPT 1987*, volume 304 of *LNCS*, pages 127–141, 1987.

[33] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In *CRYPTO 1988*, volume 403 of *LNCS*, pages 319–327. Springer-Verlag, 1988.

[34] David Chaum and Eugène van Heyst. Group signatures. In *EUROCRYPT 1991*, volume 547, pages 257–265, 1991.

[35] X. Chen, F. Zhang, D. M. Konidala, and K. Kim. New id-based threshold signature scheme from bilinear pairings. In *INDOCRYPT 2004*, volume 3348 of *LNCS*, pages 371–383. Springer- Verlag, 2004.

[36] J. D. Cohen and M. J. Fischer. A robust and verifiable cryptographyically secure election scheme. In *FOCS 1985*, pages 372–382, 1985.

[37] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO 1994*, volume 839 of *LNCS*, pages 174–187. Springer-Verlag, 1994.

[38] Ivan Damgård and Birgit Pfitzmann. Sequential iteration of interactive arguments and an efficient zero-knowledge argument for np. In *ICALP*, volume 1443 of *LNCS*, pages 772–783, 1998.

[39] Y. Desmedt and Y. Frankel. Shared generation of authenticators and signatures. In *CRYPTO 1991*, volume 576 of *LNCS*, pages 457–469. Springer-Verlag, 1991.

[40] Yvo Desmedt. Verifier-designated signatures. In *Rump Session, CRYPTO 2003*, 2003.

[41] Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In *CRYPTO 1989*, volume 435 of *LNCS*, pages 307–315. Springer-Verlag, 1989.

[42] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. IEEE *Transactions on Information Theory*, IT–22(6):644–654, 1976.

[43] Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous identification in ad hoc groups. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 609–626. Springer-Verlag, 2004.

[44] E. Egger. Considering privacy-aspects in designing cscw-applications. In *Proc. of the IFIP WG9.1 Working Conferene on NetWORKing*, pages 133–141, Vienna, Austria, June 16-18 1993. North-Holland.

[45] Uriel Feige and Adi Shamir. Zero knowledge proofs of knowledge in two rounds. In *CRYPTO 1989*, volume 435 of *LNCS*, pages 526–544. Springer-Verlag, 1989.

[46] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO 1986*, volume 263 of *LNCS*, pages 186–194. Springer-Verlag, 1986.

[47] U. Fiege, A. Fiat, and A. Shamir. Zero knowledge proofs of identity. In *STOC '87: Proceedings of the nineteenth annual ACM conference on Theory of computing*, pages 210–217, New York, NY, USA, 1987. ACM Press.

[48] B. Flinn and H. Maurer. Levels of anonymity. *Journal of Universal Computer Science*, 1(1):35–47, 1995.

[49] Y. Frankel and Y. Desmedt. Parallel reliable threshold multisignature. Technical Report TR-92-04-02, Univ. of Wisconsin–Milwaukee, 1992.

[50] Y. Frankel, P. Gemmall, P. Mackenzie, and M Yung. Proactive rsa. In *CRYPTO 1997*, volume 1294 of *LNCS*, pages 440–454. Springer-Verlag, 1997.

[51] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *CRYPTO 1997*, volume 1294 of *LNCS*, pages 16–30, 1997.

[52] Eiichiro Fujisaki and Tatsuaki Okamoto. A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In *EUROCRYPT 1998*, volume 1403 of *LNCS*, pages 32–46, 1998.

[53] Jun Furukawa and Shoko Yonezawa. Group signatures with separate and distributed authorities. In *SCN 2004*, volume 3352 of *LNCS*, pages 77–90. Springer-Verlag, 2004.

[54] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust threshold DSS signatures. In *EUROCRYPT 1996*, volume 1070 of *LNCS*, pages 354–371. Springer-Verlag, 1996.

[55] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[56] Trusted Computing Group. Trusted computing platform alliance (tcpa) main specification, version 1.1a. republished as trusted computing group (tcg) main specifcation, version 1.1b, 2001. http://www.trustedcomputinggroup.org.

[57] L. Harn. Group-oriented (t, n) threshold digital signature scheme and digital multisignature. *IEE Proc. Computers and Digital Techniques*, 141(5):307–313, 1994.

[58] P. Horster, M. Michels, and H. Petersen. Meta-multisignature schemes based on the discrete logarithm problem. In *IFIP/SEC'95*, pages 128–141. Chapman & Hall, 1995.

[59] K. Itakura and K. Nakamura. A public key cryptosystem suitable for digital multisignatures. *NEC Research & Development*, 71:1–8, 1983.

[60] M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In *EUROCRYPT 1996*, volume 1070 of *LNCS*, pages 143–154. Springer-Verlag, 1996.

[61] Aggelos Kiayias and Moti Yung. Group signatures: Provable security, efficient constructions and anonymity from trapdoor-holders. Cryptology ePrint Archive, Report 2004/076, 2004. `http://eprint.iacr.org/`.

[62] Aggelos Kiayias and Moti Yung. The vector-ballot e-voting approach. In *FC 2004*, volume 3110 of *LNCS*, pages 72–89. Springer-Verlag, 2004.

[63] Joe Kilian and Erez Petrank. Identity escrow. In *CRYPTO 1998*, volume 1462 of *LNCS*, pages 169–185. Springer-Verlag, 1998.

[64] F. Laguillaumie and D. Vergnaud. Multi-designated verifiers signatures. In *ICICS 2004*, volume 3269 of *LNCS*, pages 495–507. Springer-Verlag, 2004.

[65] Fabien Laguillaumie and Damien Vergnaud. Designated verifier signatures: Anonymity and efficient construction from any bilinear map. In *SCN 2004*, volume 3352 of *LNCS*, pages 105–119. Springer-Verlag, 2004.

[66] C. Li, M. Hwang, and N. Lee. Threshold-multisignature schemes where suspected forgery implies traceability of adversarial shareholders. In *EUROCRYPT 1994*, volume 950 of *LNCS*, pages 194–204. Springer-Verlag, 1994.

[67] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. A separable threshold ring signature scheme. In *ICISC 2003*, volume 2971 of *LNCS*, pages 12–26. Springer-Verlag, 2003.

[68] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In *ACISP 2004*, volume 3108 of *LNCS*, pages 325–335. Springer-Verlag, 2004.

[69] Joseph K. Liu and Duncan S. Wong. On the security models of (threshold) ring signature schemes. In *ICISC 2004*, LNCS. Springer-Verlag, 2005.

[70] Joseph K. Liu and Duncan S. Wong. A restricted multi-show credential system and its application on e-voting. In *ISPEC 2005*, volume ???? of *LNCS*, pages ???–??? Springer-Verlag, 2005.

[71] L.-S. Liu, C.-K. Chu, and W.-G. Tzeng. A threshold gq signature scheme. In *ACNS 2003*, volume 2846 of *LNCS*, pages 137–150. Springer-Verlag, 2003.

[72] Anna Lysyanskaya and Zulfikar Ramzan. Group blind digital signatures: A scalable solution to electronic cash. In *FC 1998*, volume 1465 of *LNCS*, pages 184–197. Springer-Verlag, 1998.

[73] Greg Maitland and Colin Boyd. Fair electronic cash based on a group signature scheme. In *ICICS 2001*, volume 2229 of *LNCS*, pages 461–465. Springer-Verlag, 2001.

[74] S. Micali, K. Ohta, and L. Reyzin. Accountable-subgroup multisignatures: extended abstract. In *CCS '01: Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 245–254. ACM Press, 2001.

[75] Atsuko Miyaji and Kozue Umeda. A fully-functional group signature scheme over only known-order group. In *ACNS 2004*, volume 3089 of *LNCS*, pages 164–179. Springer-Verlag, 2004.

[76] T. Nakanishi, T. Fujiwara, and Watanabe H. A linkable group signature and its application to secret voting. In *4th Int'l Symp. on Communicatin Theory and Appl.*, 1997.

[77] T. Nakanishi, T. Fujiwara, and Watanabe H. A linkable group signature and its application to secret voting. *Trans. of Information Processing Society of Japan*, 40(7):3085–3096, 1999.

[78] Toru Nakanishi and Yuji Sugiyama. A group signature scheme with efficient membership revocation for reasonable groups. In *ACISP 2004*, volume 3108 of *LNCS*, pages 336–347. Springer-Verlag, 2004.

[79] Moni Naor. Deniable ring authentication. In *CRYPTO 2002*, volume 2442 of *LNCS*, pages 481–498. Springer-Verlag, 2002.

[80] National Bureau of Standards FIPS Publication 46. *Data Encryption Standard*, 1977.

[81] National Bureau of Standards FIPS Publication 46-1. *Data Encryption Standard*, 1988.

[82] Lan Nguyen and Reihaneh Safavi-Naini. Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings. In *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 372–386. Springer-Verlag, 2004.

[83] National Institute of Standards and Technology (NIST). *FIPS Publication 180: Secure Hash Standard (SHS)*, May 1993.

[84] K. Ohta and T. Okamoto. A digital multisignature scheme based on the fiat-shamir scheme. In *ASIACRYPT 1991*, volume 739 of *LNCS*, pages 139–148. Springer-Verlag, 1991.

[85] K. Ohta and T. Okamoto. Multi-signature scheme secure against active insider attacks. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E82-A(1):21–31, 1999.

[86] David Pointcheval and Jacques Stern. Security proofs for signature schemes. In *EUROCRYPT 1996*, volume 1070 of *LNCS*, pages 387–398, 1996.

[87] T. Rabin. A simplified approach to threshold and proactive rsa. In *CRYPTO 1998*, volume 1462 of *LNCS*, pages 89–104. Springer-Verlag, 1998.

[88] R. Rivest. The MD4 message-digest algorithm. Internet Engineering Task Force: RFC 1320, April 1992.

[89] R. Rivest. The MD5 message-digest algorithm. Internet Engineering Task Force: RFC 1321, April 1992.

[90] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *CACM*, 21(2):120–126, Feb 1978.

[91] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565. Springer-Verlag, 2001.

[92] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[93] V. Shoup. Practical threshold signature. In *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 207–220. Springer-Verlag, 2000.

[94] Willy Susilo and Yi Mu. Non-interactive deniable ring authentication. In *ICISC 2003*, volume 2971 of *LNCS*, pages 386–401. Springer-Verlag, 2003.

[95] Isamu Teranishi, Jun Furukawa, and Kazue Sako. k-times anonymous authentication (extended abstract). In *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 308–322. Springer-Verlag, 2004.

[96] Jacques Traoré. Group signatures and their relevance to privacy-protecting off-line electronic cash systems. In *ACISP 1999*, volume 1587 of *LNCS*, pages 228–243. Springer-Verlag, 1999.

[97] Patrick P. Tsang and Victor K. Wei. Short linkable ring signatures for e-voting, e-cash and attestation. In *ISPEC 2005*, volume 3439 of *LNCS*, pages 48–60. Springer-Verlag, 2005.

[98] S. D. Warren and L. D. Brandeis. The right to privacy. *Harvard Law Review*, IV(5):193–220, 1890.

[99] A. F. Westin. Privacy and freedom. Atheneum, 1970.

[100] Duncan S. Wong, Karyin Fung, Joseph K. Liu, and Victor K. Wei. On the RS-code construction of ring signature schemes and a threshold setting of RST. In *ICICS 2003*, volume 2836 of *LNCS*, pages 34–46. Springer-Verlag, 2003.

[101] T.-C. Wu and C.-L. Hsu. Cryptanalysis of digital multisignature schemes for authenticating delegates in mobile code systems. *IEEE Transactions on Vehicular Technology*, 52(2):462 –465, 2003.

[102] S. Xu and R. Sandhu. Two efficient and provably secure schemes for server-assisted threshold signatures. In *CT-RSA 2003*, volume 2612 of *LNCS*, pages 355–372. Springer-Verlag, 2003.