Efficient group signature and identity-based signature schemes

Cui, Shi

2007

Cui, S. (2007). Efficient group signature and identity-based signature schemes. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/4194>

<https://doi.org/10.32657/10356/4194>

Nanyang Technological University

Downloaded on 26 Apr 2023 20:16:39 SGT

Efficient Group Signature and Identity-Based Signature Schemes

Cui Shi

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2007

To my parents for their encouragement.

Acknowledgments

I would like to express my sincere thanks to my supervisor, Dr. Chan Choong Wah, for his professional and outstanding guidance during my research towards the Ph.D. degree. When starting this Ph.D. program, I, a Bachelor holder with four years work experience, wondered if I could complete it in the four years. Dr. Chan's invaluable support and continuous encouragement greatly increased my confidence and made the whole research process thoroughly enjoyable. I am deeply grateful to Nanyang Technological University for funding my scholarship. Thanks also to all the people in Information System Research Lab, Information Security Center in Nanyang Technological University and Cryptography Lab in Institute for Infocomm Research (I^2R) for the nice atmosphere we have.

This work is dedicated to my family who have provided me with the ideal environment to grow up in. Their persistent emotional support always gives me the strength to pass through all kinds of difficulties. I would like to extend my special thanks to those faithful friends, for the kind help they have ever offered to me and every great time we have shared together. In particular, I would like to express my gratitude to Duan Pu and Cheng Xiangguo for insightful discussions. Others' names are not listed here but the friendship will last for long.

Abstract

This thesis introduces a group signature scheme based on RSA and some identity-based signature schemes based on the bilinear pairings over elliptic curves.

A practical group signature scheme is proposed with the introduction of a trusted third party called security mediator. The length of group signatures generated by the proposed scheme is shorter at the security level of 1024 bits than Boneh-Boyen-Shacham scheme. It is suitable for those applications which require short signatures. The implementation of the new group signature scheme is very simple and just requires modulo exponentiations.

A new identity-based signature (IBS) schemes is developed which are proven secure in the random oracle model. The security of IBS using random oracles relies on Generalized k -CAA which is an extension version of k -CAA. The experimental results show that the proposed IBS scheme in the random oracle model is the fastest one compared to the published IBS schemes from the bilinear pairings at low security levels. We also investigate methods of building non-supersingular elliptic curves suitable for pairing-based cryptosystems and find a useful property of the Frobenius trace for such curves. Based on this property, some non-supersingular elliptic curves can be built.

Batch verification is a very efficient method for speeding up multiple signa-

ture verification. We investigate batch verification of the proposed IBS scheme in the random oracle model. The experimental results exhibit excellent performance in terms of the time for verification of multiple signatures and the implementation complexity. The security analysis shows that batch verification of the proposed IBS scheme is unforgeable under an adaptive chosen message and identity attack.

Based on the proposed IBS scheme in the random oracle model, two new signature schemes with special features are developed: identity-based blind signature (IBBS) scheme and identity-based chameleon signature (IBCS) scheme. The security analysis illustrates that the new IBBS and IBCS are provably secure. The performance comparison shows that the efficiency of both new signature schemes is improved compared to the Zhang-Kim IBBS scheme and Zhang-Naini-Susilo IBCS scheme.

Contents

Acknowledgments	i
Abstract	ii
List of Figures	ix
List of Tables	x
Notations	xii
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	8
1.3 Main Contributions	11
1.4 Outline of The Thesis	12
2 Preliminaries	15
2.1 Notation	15
2.2 Provable Secure Signature Scheme	16
2.2.1 Digital Signature Scheme	16

2.2.2	Adaptive Chosen Message Attack	17
2.2.3	Random Oracle Model and Standard Model	19
2.3	Abelian Group and Finite Field	21
2.4	RSA and IFP	23
2.5	Elliptic Curves and ECDLP	24
2.6	Bilinear Pairings	26
2.6.1	Definition	27
2.6.2	Non-Rational Endomorphism	28
2.6.3	Tate pairing and Weil Pairing	28
2.6.4	Mathematical Assumptions	30
2.7	Summary	31
3	A Practical Group Signature Scheme From RSA	32
3.1	Introduction	32
3.1.1	Contribution	34
3.1.2	Prior Research	35
3.2	Definition, Security Requirements and Security Mediator	37
3.2.1	Definition	38
3.2.2	Security Requirements	39
3.2.3	Security Mediator and mRSA	41
3.3	A New Group Signature With Security Mediator	42
3.3.1	The Proposed Group Signature Scheme	44
3.3.2	Revocation	49

3.3.3	Security Analysis	50
3.3.4	Signature Size	57
3.4	Summary	59
4	Efficient and Secure Identity-Based Signature Schemes	60
4.1	Introduction	61
4.1.1	Contributions	63
4.1.2	Prior Research	64
4.2	Security Notions and Mathematical Assumptions	68
4.2.1	The Security Notion of IBS Scheme	68
4.2.2	Related Mathematical Assumption	70
4.3	An Efficient IBS Scheme in The Random Oracle Model	71
4.3.1	Generalized k -CAA	72
4.3.2	The Proposed IBS Scheme	73
4.3.3	Security Analysis of The Proposed IBS Scheme	75
4.4	A Method for Building Non-Supersingular Elliptic Curves	79
4.5	Performance	83
4.6	Summary	89
5	Applications of Identity-Based Signature Schemes	91
5.1	Introduction	91
5.1.1	Contributions	93
5.1.2	Prior Research	95
5.2	Batch verification	100

5.2.1	Definition	100
5.2.2	Batch Verification On The Proposed IBS Scheme	102
5.2.3	The Security of Type 3 Batch Verification	103
5.2.4	Performance	107
5.3	A New Identity-Based Blind Signature Scheme	109
5.3.1	Definition	109
5.3.2	The Proposed IBBS Scheme	111
5.3.3	Security Analysis	113
5.3.4	Performance	117
5.4	A New Identity-Based Chameleon Signature Scheme	120
5.4.1	Definition	120
5.4.2	Review of ZSS-IBCHF-II	122
5.4.3	The Proposed IBCS Scheme	124
5.4.4	Performance	126
5.5	Summary	127
6	Conclusion and Future Work	129
6.1	Conclusion	129
6.2	Recommendations for Further Research	132
6.2.1	Secure Short Group Signature	132
6.2.2	Batch Verification on IBS Scheme Without Random Oracles	133
6.2.3	Secure IBBS Scheme and Partial IBBS Scheme	133
6.2.4	More Identity-Based Signature Schemes	134

Author's Publications	135
Bibliography	137
A Group Law for Elliptic Curves	154
B Divisors and Miller's Algorithm	157
C Elliptic Curves Parameters	161

List of Figures

3.1	GJoin and GSign in the proposed group signature scheme	48
5.1	The blind signature issuing protocol of ZK-IBBS-I scheme . . .	112
5.2	The blind signature issuing protocol of ZK-IBBS-II scheme . .	112
5.3	The blind signature issuing protocol of the proposed IBBS scheme	112
A.1	Point addition	155
A.2	Point doubling	155

List of Tables

3.1	Comparison of the group signature length between different schemes	57
4.1	Timings of the cryptographic primitives	85
4.2	The comparison of the proposed schemes and other IBS schemes	86
4.3	Timings of the proposed schemes and other IBS schemes on $E_{ss2}^{1024}(\mathbb{F}_{512})$	86
4.4	Timings of the proposed schemes and other IBS schemes on $E_{ns2}^{1024}(\mathbb{F}_{512})$	87
4.5	Timings of the proposed schemes and other IBS schemes on $E_{ns6}^{1024}(\mathbb{F}_{157})$	87
4.6	Timings of the proposed schemes and other IBS schemes on $E_{ns4}^{2048}(\mathbb{F}_{512})$	88
4.7	Timings of the proposed schemes and other IBS schemes on $E_{ns8}^{4096}(\mathbb{F}_{512})$	89
5.1	The comparison of batch verification on the proposed scheme and other IBS schemes	107
5.2	Timings of batch verification of the proposed IBS scheme and other IBS schemes on $E_{ss2}^{1024}(\mathbb{F}_{512})$	107

5.3	The comparison of the proposed IBBS scheme and other IBBS schemes	118
5.4	Timings of the proposed IBBS scheme and other IBBS schemes on $E_{ss2}^{1024}(\mathbb{F}_{512})$	118
5.5	The comparison of the proposed IBCS scheme and ZK-IBCS schemes	127
5.6	Timings of the proposed IBCS scheme and ZK-IBCS schemes on $E_{ss2}^{1024}(\mathbb{F}_{512})$	127

Notations

- $\{0, 1\}^*$: Set of all bit strings
- 1^κ : bit string of κ ones
- $\|$: concatenation of bit string
- $\leftarrow$: assignment
- $\stackrel{R}{\leftarrow}$: randomized assignment or uniform selection
- $\equiv$: equivalence
- $Adv_{\mathcal{S}, \mathcal{A}}^{sec}(\cdot)$: advantage function of algorithm $\mathcal{A}$ in breaking scheme $\mathcal{S}$ under security notion sec
- $e_t(\cdot)$: Tate pairing
- $\hat{e}_t(\cdot)$: modified Tate pairing
- $E(K)$ or $E(\mathbb{F}_P)$: elliptic curves over finite fields K or $\mathbb{F}_P$
- φ : the distortion map
- $\mathbb{F}_q$: finite field with q elements
- $\mathbb{G}$: discrete logarithm group
- $H_i(\cdot)$: hash functions ($i = 1, 2, \dots$), possibly modelled as a random oracle
- H_M : MapToPoint hash function

- κ : security parameter
- k : embedding degree
- mod : modulo
- $\mathbb{N}$: set of natural numbers
- $\mathcal{O}$: point at infinity on an elliptic curve
- ϕ : non-rational endomorphism
- PK : public key
- $Pr[\cdot]$: probability
- SK : secret key
- $Tr(\cdot)$: trace map
- $\mathbb{Z}$: set of all integers
- $\mathbb{Z}_q$: set of integers modulo q
- $\mathbb{Z}_q^*$: multiplicative group of integers modulo q

Chapter 1

Introduction

1.1 Motivation

The rapid development of network technologies and the widespread use of the Internet make remote communication simple and allow the convenient exchange of digital information. This advancement of digital technologies has an impact on our commercial and personal activities. With so much data being transferred across the Internet, especially in the environments with low resources, the question of how to ensure the security of sensitive information becomes paramount.

In the past thirty years, digital signatures have become one of the most important areas for information security. Especially, there is a need for authentication when transferring digital information over insecure channels. Researchers have developed some regular digital signature schemes such as RSA signature scheme [151] and Digital Signature Standard (DSS) [135]. The current research work on digital signatures has gone beyond these regular digital signa-

ture schemes. Many digital signature schemes with special features have been proposed and published. For example, a blind signature scheme [56] provides specific features that make it popular when anonymity is required.

Digital signatures have proven to be a reliable technology that ensures the integrity of data, authentication and non-repudiation. They are deployed as a fundamental tool of electronic commerce all over the world. In addition, many international organizations have made digital signatures an integral component of their information security standards. These organizations include the National Institute of Standards and Technology (NIST) [135], the Institute of Electrical and Electronics Engineers (IEEE) [113] and the International Organization for Standardization (ISO) [114].

Digital signatures were first introduced by Diffie and Hellman [72]. In the paper, the authors proposed the concept of Public Key Cryptography (PKC) and announced the coming of a revolution in the cryptography community. In traditional symmetric cryptography, there is only one secret key shared by the parties of the communication. The technology required that the secret key be shared by two parties in the communication. This key had to be exchanged before a secure communication can take place. The fact that the PKC system avoids the key exchange before communication is one of PKC's most attractive features.

Under the PKC system, two different keys are required: One is called the public key and the other is called the secret(or private) key. Messages can be encrypted by using public keys or be signed with secret keys. When an encryption is carried out under a public key, only the holder of the corresponding secret key has the ability to recover the encrypted messages. In the PKC, this

scenario is called the “technology of encryption”. When a signature is generated under a secret key, anyone who knows the corresponding public key has the ability to verify it. Using the terminology of the PKC, this second scenario is associated with digital signatures. (A formal definition of digital signatures is provided in Chapter 2).

Research on digital signatures is an active branch of PKC. Since the concept of digital signature was introduced by Diffie and Hellman [72], various new schemes have been proposed. The earliest digital signature scheme, called RSA signature scheme, was based on integer factorization problem (IFP) and was proposed by Rivest, Shamir and Adleman [151]. This was soon followed by the ElGamal [80] signature scheme. This scheme was based on discrete logarithm problem (DLP). Vanstone [166] proposed elliptic curve digital signature algorithm (ECDSA), based on the elliptic curve discrete logarithm problem (ECDLP). These three schemes are now accepted as the regular signature schemes because they can facilitate the regular services of data integrity, authentication and non-repudiation. Other well known regular digital signature schemes include the Rabin signature [150], Schnorr signature [155] and Okamoto signature [141]. Besides the mentioned regular signature schemes, there are a lot of signature schemes with special features, such as blind signature [56], group signature [61] and so on. A brief review about these signature schemes with special features will given in the end of this section.

So far, it is still an active area to propose new digital signature based on new mathematical assumptions and construct digital signature schemes with new special features. However, there exists some problems or challenges when those defined regular signatures or signature with special features are deployed

in the low-resource environments. One problem is how to reduce or avoid the transport delay caused by the long signature. In wireless networks, signatures are often contained in the transported information between mobile devices and wireless network devices, and short signature can reduce the transport delay. The other problem is how to reduce power consuming caused by public key cryptography. The arithmetics of the public key cryptography requires large amounts of processing power. Reducing arithmetics of public key cryptography or improving the efficiency of signing messages and verifying signatures is of benefit to those low power consumer devices, such as cell phones or PDA. The above description focuses on the same problem, that is, how to improve the performance of the signature schemes. The motivation of this thesis is to enhance the performance of the signature schemes with/without special features such that they are deployed in the environments with low resource.

A brief survey of digital signatures: The signature schemes with special features can provide some properties that the regular signatures cannot provide, e. g. anonymity provide by blind signature and group signature. We give a simple literature review of these digital signatures with special features so as to understand which signatures satisfy which environments. The following are a brief review of some signatures that incorporate special features:

- *Blind signature* by Chaum [56] is used to disguise the content of a message before it is signed. The resulting blind signature can be publicly verified against the original, unblinded message in a manner similar to a regular digital signature.
- *Multi-signature* by Itakura and Nakamura [116] allows any subgroup of a

group of players to jointly sign a message such that a verifier is convinced that each member of the subgroup participated in the signing.

- *Identity-based (ID-based) signature* by Shamir [161] is similar to regular digital signature schemes except that the public key holds information on the signer's identity such as his or her name, email address or IP address. This signature is useful because it simplifies the key management system.
- *Undeniable signature* by Chaum and Antwerpen [59] allows the signer to set limits as to who can verify their signatures. If the encrypted information were to reach a third party, they would not be able to verify the validity of the signature.
- *One-time signature* introduced by Lamport [117] allows the signature of only a single message using a given piece of private (and public) information so as to obtain fast performance.
- *On-line/off-line signature* by Even, Goldreich and Micali [81] is created by combining a regular signature scheme with a one-time scheme. It allows the bulk of the computational burden of a signature operation to be performed even before the signer knows the message that will be signed.
- A (t, n) -*threshold signature* by Desmedt and Frankel [71] allows a group of n players to generate signatures on a message by sharing a secret key. Any subgroup of more than t ($t < n$) players can then use their parts of the secret key to generate a signature that can be verified by anyone with the unique fixed public key.
- *Group signature* by Chaum and van Heyst [61] allows a member of a group to sign a message on behalf of the entire group. The verifier, in this case,

can confirm that the message came from the group, but he has no way of finding out which member in the group actually signed the message.

- *Designated confirmer signature* by Chaum [58] allows certain designated parties to confirm the authenticity of a message without requiring any input from the signer. At the same time however, it is not possible to verify the authenticity of a given message without the aid of either the signer or the designated parties.
- *Proxy signature* by Mambo, Usuda and Okamoto [123] allows a proxy to sign the message on behalf of the original signer. The proxy's signature can be classified as full delegation, partial delegation, and delegation by a warrant.
- *Signcryption* scheme by Zheng [184] is designed to both digitally sign and encrypt a message. The fact that the processes are conducted simultaneously means that authenticity and confidentiality are assured in a shorter time than in traditional systems.
- *Chameleon signature* by Krawczyk and Rabin [110] provides a signature for a designated recipient which cannot be validated by another party. If presented with a false signature claim, the signer can prove that the signature is forged.
- *Ring signature* by Rivest, Shamir and Tauman [152] is a simplified group signature with no manager, no group setup procedure, and no revocation mechanism. It assures the signer's anonymity and lets the verifier only know that the signature is generated from a fellow ring member.

- *Transitive signature* by Micali and Rivest [127] is used to sign the vertices and edges of a directed or undirected finite graph. A transitive signature scheme is compatible with computing the transitive closure of the graph being signed.
- *Aggregate signature* by Boneh et al. [38] supports the aggregation of many signatures. Namely, given n signatures on n distinct messages from n distinct users, all these signatures can be aggregated into a single short signature such that this single signature convinces the verifier that the n users did indeed sign the n original messages.
- *Verifiably encrypted signature* by Asokan et al. [5] is used in contract-signing protocols which enables the verifier to test that a given ciphertext C is the encryption of a signature on a given message m . The verifiably encrypted signature scheme based on the bilinear pairing were proposed in Boneh et al. [38].

In addition to the list above, combinations of two or more digital signatures have been created in order to deliver more efficient features. For example, the *identity-based blind signature* by Zhang and Kim [178] allows identity information to be used as public keys when signing blind messages. A (t, n) -*blind threshold signature* by Juang and Lei [104] allows any t out of n signers in a group to represent the group by signing blind threshold signatures. A (t, n) -*threshold proxy signature* by Zhang [176] allows any t or more proxy signers to sign a message on behalf of an original signer for the specified verifiers. A *group blind signature* by Lysyanskaya and Ramzan [122] allows a group member to sign a message on behalf of the group without knowing the content of the message. Refer to [120] and [169] for more information on digital signature

schemes.

1.2 Objectives

The current problems are how to minimize the transport delay caused by long signature and how to minimize the power consuming caused by the arithmetics of the public key cryptography. The main objective of this thesis is to solve the current problem. To achieve this objective, this thesis concentrates on proposing a group signature scheme with short signature and three efficient identity-based signature schemes. The main objective includes four sub-objectives which are given as follows:

The group signature, introduced by Chaum and van Heyst [61], is very useful in applications when signing messages requiring anonymity. Each member of the group can sign a message on behalf of the group, but the identity of the signer is kept secret. No one, except the manager of the group, can identify the actual signer. Recent applications which need group signatures have emerged [34]. However, in these applications, the requirements for group signatures are more rigorous. Not only is efficient implementation necessary, the size of the group signature also has to be short so as to reduce or avoid the caused transport delay. In line with this demand, researchers attempt to construct new group signature schemes so as to avoid long group signatures. *The first sub-objective is to introduce a practical group signature scheme which provides short group signatures.*

Identity-based encryption (IBE) has been an open problem since the concept of identity-based Public Key Cryptography (ID-PKC) was proposed by Shamir

[161] in 1984. It was not solved until 2001. Boneh and Franklin [36] successfully constructed a practical identity-based encryption (IBE) scheme by using bilinear pairings and Cocks [69] designed an IBE scheme based on the Quadratic Residuosity Problem. Since then, ID-PKC has become an attractive research area in the cryptography community. A large amount of identity-based cryptographic schemes have been proposed based on the bilinear pairings. ID-PKC is viewed as an alternative to PKC in situations where the security requirements are not too rigorous. Similar to traditional PKC, there is also a public/secret key pair in ID-PKC. The only difference is that the public key in ID-PKC is the identity information of the owner of the key pair, such as personal name, identity card number, IP address and names of companies or organizations. This helps to establish a relationship between the owner and the public/secret key pair. In traditional PKC, the relationship between the owner and the public/secret key pair is maintained by a so-called certificate, which is used to bind the identity of the owner of the key pair and the corresponding key pair together. Certificates are issued by a third party, called the certification authority (CA). When the number of users is very large, CA has to maintain a large database for certificates storage and keys management. Because the public key in ID-PKC shows the identity of the owner of the key pair, certificates and the corresponding space for certificate storage are not needed. Key management is thus simplified. This makes ID-PKC advantageous over traditional PKC. For this reason, researchers pay more attention to identity-based cryptographic schemes after IBE problem was solved. Although Shamir [161] offered an identity-based signature (IBS) scheme based on IFP when introducing the concept of ID-PKC, his scheme is inefficient in real applications. Recently, some excellent IBS schemes [153, 142, 54, 65, 100, 174, 175] from the bilinear pairings were constructed.

However, to deploy identity-based signature schemes in the environments with low resource and the devices with low power consumer, further research work on fast signing message and verifying signatures are expected from the community. *In light of this expectation, the second sub-objective is to introduce an identity-based signature scheme in the random oracle model which is more efficient than the current IBS schemes [153, 142, 54, 65, 100, 174, 175].*

It has been noted that the technology of batch verification is supported by many IBS schemes. This enables many signatures to be verified simultaneously and hence shortens the verification time. In practice, it is desired that IBS schemes support batch verification for any signatures that are used on different messages by one user or on different messages by different users. *The third sub-objective is to introduce batch verification of the proposed IBS scheme in the random oracle model, which is faster than the IBS scheme by Yoon, Cheon and Kim [175].*

Constructing identity-based digital signature schemes with special features has been an active research field during the last five years or so. The functions of identity-based signature schemes are similar to their corresponding counterparts in traditional PKC, although in the former the public key consists of the user identity information, such as personal name, email address or IP address. Identity-based blind signature (IBBS) and identity-based chameleon signature (IBCS) are two identity-based digital signatures with special functions. It should be noted that IBBS schemes or IBCS schemes do not perform well because the IBS schemes used in them are inefficient. To avoid this, it is desired that the IBS scheme should be as fast as possible so that an IBBS or IBCS will be able to offer very efficient performance. *The fourth sub-objective*

is to investigate the use of the proposed IBS scheme in the random oracle model in an identity-based blind signature and an identity-based chameleon signature, and prove that they can achieve better performance.

1.3 Main Contributions

The contribution of the research covers a series of optimized digital signature schemes with short signature size, efficient performance and provable security.

Specific research contributions of this thesis include:

- A new group signature scheme has been proposed. With the aid of a trusted online third party, called a *security mediator*, the size of the group signature is reduced to be the same as a standard RSA signature with the same security level. Although there is no formal security model for proving the security of the proposed group signature scheme, a informal security analysis show that the proposed group signature scheme meets all the requirements of a secure group signature.
- A new IBS scheme in the random oracle model has been proposed based on the bilinear pairings over elliptic curves. The security analysis shows that their security is provable and depends on different mathematical assumptions. Additionally, an alternative method of finding non-supersingular elliptic curves suitable for pairing-based cryptosystems has been introduced. The proposed IBS scheme has been implemented on supersingular and non-supersingular elliptic curves. The experimental results validate the conclusion that the performance of the proposed IBS scheme is better than other IBS schemes.

- Two types of batch verification of the proposed IBS scheme in the random oracle model have been developed. Both types of batch verification can verify multiple signatures simultaneously, which can significantly reduce the time taken to verify signatures, while not compromising the security of the original IBS scheme. The experimental results show that batch verification exhibits excellent computational efficiency.
- The proposed IBS scheme in the random oracle model has been used to construct a new IBBS scheme. The experimental results show that the new IBBS scheme provides efficient performance in a blind signature issuing protocol and requires less storage space for public parameters. The security analysis shows that the proposed IBBS scheme meets all the requirements of a secure IBBS scheme.
- An efficient identity-based chameleon signature (IBCS) has been constructed based on the combination of an existing identity-based chameleon hash function and the proposed IBS scheme in the random oracle model. The key generation for both the identity-based chameleon hash function and the proposed scheme is unified. The new IBCS scheme is more efficient than IBCS scheme based on the current IBS schemes.

1.4 Outline of The Thesis

In this thesis, we focus on the improvement and design of digital signature schemes. The outline of this thesis is as follows.

Chapter 2 introduces the formal definition of digital signatures as well as some fundamentals of secure digital signature schemes. It then presents a few

classical mathematical problems used in Public Key Cryptography such as integer factoring problem (IFP) and elliptic curve discrete logarithm problem (ECDLP). This chapter also introduces the definition of bilinear pairings on elliptic curves.

In Chapter 3, a practical group signature scheme is designed with the help of a trusted third party called the *security mediator*. The proposed group signature scheme exhibits better performance in terms of the size of the group signature.

Chapter 4 is devoted to the construction of IBS scheme with high efficiency in the random oracle model. An efficient IBS scheme is developed in the random oracle model. Based on a new complexity assumption, the proposed IBS scheme in the random oracle model is shown to be provably secure. The proposed IBS scheme is from the bilinear pairings over elliptic curves. Some experimental results are obtained when the proposed IBS scheme is implemented on supersingular and non-supersingular elliptic curves with different embedding degrees. Furthermore, a simple method for building suitable non-supersingular elliptic curves is introduced in this chapter.

In Chapter 5, we investigate the use of identity-based digital signatures in different applications. Based on the proposed IBS scheme in the random oracle model, a method of verifying multiple signatures simultaneously, an efficient identity-based blind signature (IBBS) scheme and an efficient identity-based chameleon signature (IBCS) scheme are developed. These schemes (or methods) provide better performance than their corresponding counterparts in the literature, and their improvement does not compromise their security.

In the three chapters, the theoretical analysis of the performance of all the proposed digital signature schemes is illustrated and is compared with the cor-

responding counterparts. Experimental results on different cases are also presented to support our theoretical results.

Finally, in Chapter 6, some concluding remarks are given and possible directions for future work are discussed.

Chapter 2

Preliminaries

2.1 Notation

Let $\{0, 1\}$ denote the set of individual bits and let $\{0, 1\}^*$ denote the set of all bit strings. $\mathbb{N} = \{0, 1, 2, \dots\}$ is the set of all natural numbers. $\kappa \in \mathbb{N}$, 1^κ and $\{0, 1\}^\kappa$ denotes the bit string of κ ones and the set of bit strings of length κ , respectively. When a and b are strings, then $|a|$ refers to the length of the string a and $a \parallel b$ indicates the concatenation of a and b . When S is a set, $|S|$ is its cardinality. $x \xleftarrow{R} S$ or $x \in_R S$ denotes that an element x is chosen randomly from the set S .

2.2 Provable Secure Signature Scheme

2.2.1 Digital Signature Scheme

Digital signatures are one of the most important cryptographic tools in modern cryptography. In Public Key Cryptography (PKC), every user has two keys: One is the *public key* which is published to everyone, and the other is the *secret (or private) key* which is kept secret by the corresponding owner. In a traditional PKC, the key pair is generated by users or by a key generator called Certificate Authority (CA). In identity-based PKC (ID-PKC), the key pair is generated by a key generator called Trusted Authority (TA). It is worthy of noting that their roles are different. When the key pairs are generated by users, the CA does not know the users' secret key and just issues the certificate with the binding of identities and public keys. The TA checks that applicants have claimed identity and then issues the corresponding secret key. Thus ID-PKC have a key escrow facility. Whether this facility is useful or not depends on the particular applications. Both PKC and ID-PKC use the difficulty of the mathematical problems, such as integer factoring problems and discrete logarithm problems, to guarantee that no one is able to recover the secret (private) key from the public key. The difference between PKC and ID-PKC is that the public key is random in PKC, while the public key in ID-PKC is specified before generating the private/secret key.

The signature attached to a message generated by a user is a string, with its value depending on the data in the message, the public key, the user-specific secret key, and other randomly chosen elements. Anyone who knows the public key can check the validity of the signatures. Digital signatures thus provide

messages with the crucial attributes of data integrity and non-repudiation. The definition of a regular signature scheme is as follows [124]:

Definition 2.1. *A regular digital signature scheme includes three algorithms: the key generation algorithm (**KeyGen**), the signing algorithm (**Sign**) and the verification algorithm (**Verify**).*

- **KeyGen**: *This is a probabilistic algorithm. Given a security parameter κ , on input 1^κ , the algorithm **KeyGen** produces a pair (K_p, K_s) . The user keeps the secret key K_s and publishes the public key K_p .*
- **Sign**: *This is a probabilistic algorithm. On inputting a message m , the user's secret key K_s and possibly some random elements, the algorithm **Sign** will produce a signature σ . We denote this as $\sigma = \text{Sign}(m, K_s)$.*
- **Verify**: *This is a deterministic algorithm. Upon entering a signature σ , a message m and a public key K_p , the algorithm **Verify** will return a boolean value by checking the validity of the signature σ . We denote this as $\text{Verify}(m, K_p, \sigma)$. If $\text{Verify}(m, K_p, \sigma) = \text{True}$, then the verifier accepts the signature. Otherwise, the signature will be rejected.*

2.2.2 Adaptive Chosen Message Attack

There are two types of attacks against regular digital signatures: One is the *no-message attack* and the other is *known-message attack*. Attackers using the no-message attack only know the public key of the signer, while attackers using the known-message attack have access to a list of message-signature pairs. Known-message attacks can be classified as four different sub-attacks: *the plain known-message attack*, *the generic chosen-message attack*, *the oriented chosen-message*

attack and *the adaptively chosen-message attack*. So far, the most effective attack against digital signature schemes is the adaptive chosen message attack. Each of the above sub-attacks has the following three expected results:

- *Total break*: attackers obtain the secret key of the signer;
- *Universal forgery*: attackers are able to construct an algorithm to sign any message and obtain valid signatures;
- *Existential forgery*: attackers are able to provide a new valid message-signature pair. [149]

The danger of total break and universal forgery is much larger than that of existential forgery. Pointcheval and Stern [149] define a secure regular digital signature as: A signature scheme is secure if no algorithm can generate an existential forgery under an adaptive chosen-message attack. The strongest notion of security for regular signature schemes was defined by Goldwasser et al. [95] in this way:

Definition 2.2. *A signature scheme $S = \langle \text{KeyGen}, \text{Sign}, \text{Verify} \rangle$ is existentially unforgeable under an adaptive chosen message attack if it is infeasible for a forger to produce a valid message-signature pair, when the forger knows only the public key and possesses multiple signatures from the signer on messages of the forger's choice.*

Let $\text{Adv}(\mathcal{F})$ be the advantage of $\mathcal{F}$. Formally, for every probabilistic polynomial time forger algorithm, $\mathcal{F}$, there does not exist a non-negligible probability

ϵ such that

$$\text{Adv}(\mathcal{F}) = \Pr \left[\begin{array}{l} (K_p, K_s) \leftarrow \text{KeyGen}(1^\kappa) \\ \text{for } i = 1, 2, \dots, \kappa; \\ m_i \leftarrow \mathcal{F}(K_p, m_1, \sigma_1, \dots, m_{i-1}, \sigma_{i-1}), \sigma_i \leftarrow \text{Sign}(K_s, m_i); \\ (m, \sigma) \leftarrow \mathcal{F}((K_p, m_1, \sigma_1, \dots, m_\kappa, \sigma_\kappa); \\ m \notin \{m_1, m_2, \dots, m_\kappa\} \text{ and } \text{Verify}(K_p, m, \sigma) = \text{True} \end{array} \right] \geq \epsilon$$

where $\kappa \in \mathbb{N}$ is the security parameter.

2.2.3 Random Oracle Model and Standard Model

The Random Oracle Model: The hash function is a very useful tool in modern cryptography because it has the property of collision-resistant. Collision-resistant means that it is impossible for anyone to find two different messages providing the same hash value. The hash function was originally designed to be used in digital signature schemes for signing long messages by reducing the size of the signatures without comprising the integrity of the message. In order to obtain related arguments in the security proof of cryptographic schemes, some cryptographers [85, 21, 23, 146, 147, 148] have suggested that hash functions should be seen as random functions. Therefore, a new model, the *random oracle model* (ROM), has been introduced into the security proofs of cryptographic schemes. If one expects to prove the security of a cryptographic scheme where hash functions are used, then these hash functions are viewed as random oracles. This often makes it easy and simple to prove the security of cryptographic schemes. The random oracles will return a truly random value for each new query from a challenger. If a new query is the same as a previous query, the

random oracle will return a identical answer. This theoretical model can help cryptographers verify whether the design of a cryptographic scheme is secure or not. However, the random oracle model is only an imaginary model in theory. There does not exist a true random oracle in the real world. In real world applications, hash functions such as HAVAL [145], SHA-224 [136] or RIPEMD-160 [75], play the role of random oracles. Since the concept of the random oracle model was introduced into the community, researchers have used it to provide the security proof for numerous cryptographic schemes.

The Standard Model: Although many cryptographic schemes have been proven in the random oracle model, some researchers doubted the reliability of this model. As the description in the previous section, hash functions are usually treated as random imaginary functions when proving the security of the cryptographic schemes in the random oracle model¹. Recently, Canetti, Goldreich and Halevi [52, 53] investigated the reliability of the random oracle and presented some useful results. They showed that there were some examples of cryptographic schemes which are secure under the random oracle model but not in real applications. Therefore, constructing cryptographic schemes without using random oracles, namely in the standard model, has become an active research area. Encryption schemes and signature schemes that do not use random oracles have been proposed [31, 32, 33, 172]. Their security has been proven in the standard model. It is noted that, although the authors of [53] have shown the weakness of the random oracle model, they did not agree that the random oracle model was of no use in proving the security of the cryptographic schemes. This is a “confusing” conclusion. In addition, Mao has commented that [124]:

¹It is worthy of noting that not all the schemes using hash functions prove their security by the random oracle. Those which only require the collision resistance property can be proven secure in the standard model, such as Cramer-Shoup cryptosystem [70].

“The ROM-based technique for security proof manifests its importance in that it suggests where to focus the attention for careful design”. Until now, there has been no explicit conclusion that cryptographic schemes should not use random oracles. Therefore, new cryptographic schemes with random oracles whose security is proven in the random oracle model are still being constructed.

2.3 Abelian Group and Finite Field

Abelian groups and finite fields are two important algebraic structures in modern cryptography. They are widely used in real world applications because they provide the basic elements and operations for cryptographic schemes.

A group is thought as a set of objects with an operation between two elements in the set. It can be described by mathematical language as follows [124]:

Definition 2.3. *A group $(G, \diamond)$ is a set G together with an operation $\diamond$ which satisfies the following conditions:*

- *For all $a, b \in G$, $a \diamond b \in G$.*
- *For all $a, b, c \in G$, $a \diamond (b \diamond c) = (a \diamond b) \diamond c$.*
- *There exists a unique element $d \in G$, for all $a \in G$, $a \diamond d = d \diamond a = a$. This element d is called the identity element.*
- *For all $a \in G$, there exists a^{-1} such that $a \diamond a^{-1} = d$.*

There is only one operation (or group law) in a group. If the operation is the addition, then the group is an additive group and is denoted by $(G, +)$. In an additive group, the identity element is called the additive identity and

is denoted by 0. If the operation is the multiplication, then the group is a multiplicative group and is denoted by $(G, \cdot)$. In a multiplicative group, the identity element is the multiplicative identity and is denoted by 1. For example, the set of points on an elliptic curve over a finite field is an additive group where the group law is addition. If $n = p \cdot q$ for different prime integers p, q , then the non-zero elements of $\mathbb{Z}$ modulo n under multiplication form a multiplicative group which is denoted by $\mathbb{Z}_n^*$.

We usually omit the operation $\diamond$ and use G directly to represent a group. In a group G , if $a \diamond b = b \diamond a$ for all $a, b \in G$, then this group is called an *abelian group*. If the number of elements in a group G is finite, this group is a *finite group*. The number of elements in a finite group is called the order of G and is denoted by $|G|$ or $\#G$.

A ring R is a set with two operations. Its definition is as follows [124]:

Definition 2.4. *A ring is a set together with an addition operation $+$ and a multiplication operation $\cdot$ which satisfies the following properties*

- For all $a, b \in R$, $a \cdot b = b \cdot a$;
- For all $a, b, c \in R$, $a \cdot (b + c) = a \cdot b + a \cdot c$.

Next, we introduce the definitions of the field and the finite field [124].

Definition 2.5. *If the non-zero elements of a ring form a group under multiplication, then the ring is called a field and is denoted by $\mathbb{F}$. If a field has a finite number q of elements in it (where $q \in \mathbb{Z}$), then this field is called a finite field and is denoted by $\mathbb{F}_q$.*

In modern cryptography, q is a prime or a power of a prime for all the finite fields. When $q = p$ is a prime number, the set of integers modulo p forms a

finite field with addition and multiplication performed modulo p . This finite field $\{1, 2, \dots, p-1\}$ is called a prime field and is denoted by $\mathbb{F}_p$.

2.4 RSA and IFP

RSA signature was proposed by three well known computer scientists, Rivest, Shamir and Adleman [151], in 1978. It was the first digital signature scheme introduced to the community. The security of RSA signatures depends on the so-called RSA problem [124]. Let p and q be two large prime integers and $n = p \cdot q$ where $|n| \geq 1024$. Let $\phi(n) = (p-1)(q-1)$ denote *Euler's phi function* of n (see Definition 5.11 in [124]). RSA problem is defined as follows [124]:

Definition 2.6. *Let n be a product of two primes p and q and e be an integer such that $\gcd(e, \phi(n)) = 1$. Given $c \in \mathbb{Z}_n^*$, the RSA problem asks to find $d \in \mathbb{Z}_n^*$ which satisfies $d^e \equiv c \pmod{n}$.*

RSA problem is defined as the task of computing the e -th roots modulo of a composite n . The difficulty of RSA problem relies on another famous mathematical assumption, the integer factorization problem (IFP). Namely, given n as described above and no knowledge of p and q , it is computationally infeasible to factor n into its prime factors. Thus far, nobody has found a polynomial time algorithm for solving IFP on a classical computer. However, this does not mean that such a polynomial time algorithm does not exist.

2.5 Elliptic Curves and ECDLP

Elliptic curve cryptography (ECC) was proposed by Koblitz [108] and Miller [128]. They independently suggested the use of elliptic curve groups in Public Key Cryptography. Finite abelian groups are fundamental to Public Key Cryptography. Elliptic curves over finite fields offer a large number of such finite abelian groups. In the past twenty years, researchers have paid much attention to ECC and developed many important cryptographic schemes. Compared with cryptosystems based on RSA or DLP, ECC has several advantages. One of them is that ECC is able to provide reliable security with shorter key lengths than other cryptosystems.

Next, we review the definition of elliptic curves over finite fields. Let $K = \mathbb{F}_p$ be a finite field. Suppose $E(K)$ is an elliptic curve E over a field K , it is defined as the set of points (x, y) with $x, y \in K$ that satisfy the Weierstrass equation:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (2.1)$$

where $a_1, a_2, a_3, a_4, a_6 \in K$. $E(K)$ also includes an extra point $\mathcal{O}$ called *point at infinity* which is usually denoted by $\mathcal{O} = (x, \infty)$. The number of points on $E(K)$ is called the order of $E(K)$ which is denoted by $|E(K)|$ or $\#E$. When $K = \mathbb{F}_p$ for a large prime $p \geq 5$, the Weierstrass equation shown in Equation 2.1 is simplified into the following equation

$$y^2 = x^3 + a_4x + a_6 \quad (2.2)$$

For supersingular elliptic curves (see Definition 2.8), if $K = \mathbb{F}_p$ and $p = 2^m$, the

Weierstrass equation (Equation 2.1) is simplified into the following equation

$$y^2 + a_3y = x^3 + a_4x + a_6 \quad (2.3)$$

For ordinary (non-supersingular) elliptic curves (see Definition 2.8), if $K = \mathbb{F}_p$ and $p = 2^m$, the Weierstrass equation (Equation 2.1) is simplified into the following equation

$$y^2 + a_1xy = x^3 + a_2x^2 + a_6 \quad (2.4)$$

The points on $E(K)$ form an abelian group under an addition operation. Let P, Q be two points on $E(K)$. Point addition, $P + Q$, refers to computing the sum of these two points. See Appendix A for more details on the group law on $E(\mathbb{F}_p)$ and $E(\mathbb{F}_{2^m})$. If λ is an integer, λP indicates point multiplication (or scalar multiplication), which refers to the execution time of cryptographic schemes based on elliptic curves. Increasing the speed of point multiplication is an interesting research topic in ECC. [99] provides a detailed survey of point multiplication. ECC depends on the difficulty of *the elliptic curve discrete logarithm problem* (ECDLP), which is the application of the discrete logarithm problem (DLP) to elliptic curve groups. Its definition is as follows:

Definition 2.7. *For an elliptic curve E over a finite field K , where P, Q are two points on $E(K)$, it is difficult to determine $\lambda \in \mathbb{Z}$ such that $Q = \lambda P$.*

Menezes, Okamoto and Vanstone [125] proposed an efficient attack method for supersingular elliptic curves called MOV attack. That is, an ECDLP on supersingular elliptic curves can be reduced to a much easier DLP in a finite field by using bilinear pairings, such as [125] or Tate pairing [86]. Therefore,

such a supersingular elliptic curve was seen as weak and to be avoided when selecting elliptic curve parameters. Its definition is as follows:

Definition 2.8. *Suppose that $E(\mathbb{F}_p)$ is an elliptic curve defined over a finite field $\mathbb{F}_p$ and q is the characteristic of $\mathbb{F}_p$. $E(\mathbb{F}_p)$ is supersingular if q divides t where t is the trace. If q does not divide t , then $E(\mathbb{F}_p)$ is non-supersingular or ordinary.*

Although there was limited discussion on supersingular elliptic curves in early literature, the situation is different today. Researchers have found that bilinear pairings on supersingular elliptic curves can be used to construct positive cryptographic schemes with speed and efficiency.

2.6 Bilinear Pairings

The early positive applications based on the bilinear pairings include Sakai et al.'s signature scheme [153] and Joux's three-party key agreement protocol [103]. Boneh and Franklin [36] used the bilinear pairings to construct an identity-based encryption (IBE) scheme, which was a well-known open problem over the past fifteen years. Since then, numerous cryptographic schemes based on the bilinear pairings have been constructed (see website [12]). These schemes from the bilinear pairings form a branch of ECC called pairing-based cryptosystems (PBC). This new branch includes the construction of cryptographic schemes from the bilinear pairings [36, 15], improving the efficiency of computing pairings algorithms [13, 88] and building suitable elliptic curve parameters [16, 78]. In this section, we introduce some preliminary knowledge of bilinear pairings.

2.6.1 Definition

Let q be a positive integer. G_1 and G_2 are two additive groups of order q where DLP is difficult. G_3 is a multiplicative group of order q . A bilinear pairing e can be defined as:

$$e : G_1 \times G_2 \longrightarrow G_3$$

Currently, the bilinear pairings used in most existing cryptographic schemes are derived from Tate pairings or Weil pairings, which can be implemented on a supersingular elliptic curve or a suitable non-supersingular elliptic curve. If there is no special explanation in this thesis, G_1 and G_2 are regarded as two Abelian groups constructed by elliptic curves over finite fields². Let k be an integer larger than 1. In the following description, K is either a finite field $\mathbb{F}_{p^k}$ or a subfield of $\mathbb{F}_{p^k}$, and k is termed an embedding degree or security parameter. The bilinear pairings in ECC are usually thought of as functions which map two elliptic curve points to an element of a multiplicative group. Let $\mathcal{O}$ denote the identity element of G_1 or G_2 and let 1 denote the identity element of G_3 . The bilinear pairings have the following properties:

- Bilinearity: For any $P_1, P_2 \in G_1, Q_1, Q_2 \in G_2$, $e(P_1 + P_2, Q_1) = e(P_1, Q_1) \cdot e(P_2, Q_1)$ and $e(P_1, Q_1 + Q_2) = e(P_1, Q_1) \cdot e(P_1, Q_2)$.
- Non-degeneracy: For all $P \in G_1$ except $\mathcal{O}$, there is some point $Q \in G_2$ such that $e(P, Q) \neq 1$. In the same way, for all $Q \in G_2$ except $\mathcal{O}$, there is some point $P \in G_1$ such that $e(P, Q) \neq 1$.

²Abelian groups for bilinear pairings are constructed not only by the points of elliptic curves over finite fields; other groups, such as the points of the Jacobian of hyperelliptic curves, are also used to construct Abelian groups.

2.6.2 Non-Rational Endomorphism

Non-rational endomorphism, denoted by ψ , is very useful in pairing-based cryptosystems. It is used to convert the points between G_1 and G_2 . There are two important non-rational endomorphisms: the distortion map [167] and the trace map [40]. It is noted that the distortion map works only on supersingular elliptic curves, while the trace map works on both supersingular elliptic curves and non-supersingular elliptic curves. The following briefly introduces their definitions. For further details, see Chapter IX of [29].

Distortion Map: The distortion map is used to map a point from $E(\mathbb{F}_p)$ to $E(\mathbb{F}_{p^k})$. Given a supersingular elliptic curve, there exists at least one distortion map. However, for any non-supersingular elliptic curve, no valid distortion map exists. We denote the distortion map by φ : $Q = \varphi(P)$ where $P \in E(\mathbb{F}_p)$ and $Q \in E(\mathbb{F}_{p^k})$.

The Trace Map: The trace map is used to map a point from $E(\mathbb{F}_{p^k})$ to $E(\mathbb{F}_p)$. Suppose that P is a point with the coordinates (x_P, y_P) on the elliptic curve $E(\mathbb{F}_{p^k})$. The trace map is then defined as $Tr(P)$: $Tr(P) = \sum_{i=0}^{k-1} (x_P^{p^i}, y_P^{p^i})$.

2.6.3 Tate pairing and Weil Pairing

Tate pairings and Weil pairings are two bilinear pairings widely used in PBC. Their definitions and properties are described below:

Suppose $G_1 = E(\mathbb{F}_p)$ and $G_2 = E(\mathbb{F}_{p^k})$, $|G_1| = |G_2| = q$. $P \in G_1$ and $Q \in G_2$. Suppose f_P is a non-zero function on $E(\mathbb{F}_p)$ (See Appendix B). Its divisor (f_P) is represented by: $(f_P) = q(P) - q(\mathcal{O}) \in \mathbb{F}_{p^k}^* / (\mathbb{F}_{p^k}^*)^q$. Tate pairing³ is

³Tate pairing is originally defined by $e_t(P, Q) = f_P(Q)$. For computational convenience, $e_t(P, Q)$ are raised to the power $(p^k - 1)/q$ so as to eliminate all q -th power and leave an exact

defined by

$$e_t(P, Q) = f_P(Q)^{(p^k-1)/q} \subseteq \mathbb{F}_{p^k}^*$$

Suppose $E[q]$ is a group of q -torsion points [171] on the elliptic curve E and $G_1 = G_2 = E[q]$. $P, Q \in G_1$. D and D' are two different degree zero divisors such that the support of D and D' are disjoint. Suppose $D \sim (P) - (\mathcal{O})$ and $D' \sim (Q) - (\mathcal{O})$. f and g are two non-rational functions such that $(f) = qD$ and $(g) = qD'$. Weil pairing is defined by

$$e_w(P, Q) = \frac{f(D')}{g(D)} \in G_3$$

where $G_3 = \mathbb{F}_{p^k}^*$.

When designing cryptographic schemes based on the bilinear pairings, modified pairings [29] are more convenient as they usually use a single subgroup G_1 of $E(\mathbb{F}_p)$, with a prime order q and generator P as inputs. Let $P, Q \in E(\mathbb{F}_p)$. φ denotes the distortion map. A modified Tate pairing $\hat{e}_t$ on a supersingular elliptic curve can be denoted by:

$$\hat{e}_t(P, Q) = e_t(P, \phi(Q))$$

A modified Tate pairing on a supersingular elliptic curve has two very useful properties:

- $\hat{e}_t(P, P) \neq 1$ for all $P \in G_1$.
- Symmetry: For any $P, Q \in G_1$, $\hat{e}_t(P, Q) = \hat{e}_t(Q, P)$.

n -th root of unity in $\mathbb{F}_{p^k}$.

The first property is very important. Certain cryptographic schemes that use bilinear pairings often require this property. The second property helps P and Q to interchange flexibly during the computation of pairings, enabling researchers to easily design cryptographic schemes. Due to this property, modified Tate pairings on supersingular elliptic curves are also called symmetric pairings. However, modified Tate pairings on non-supersingular elliptic curves do not have the second property.

The research on the bilinear pairings is just at the beginning stage, and there are some unsure conclusions. The efficiency of computing a pairing at different security levels is one of them. Tate pairing is thought to be more efficient than Weil pairing. Previous research in [109, 157] showed that Tate pairing is more efficient than Weil pairing only when the embedding degree is small. If the embedding degree is a large number, computing a Weil pairing is more efficient than computing a Tate pairing. However, recent research by Granger et al. [96] offers a new method of analysis, such that Tate pairing provides more efficient performance even in the case of a large embedding degree.

2.6.4 Mathematical Assumptions

Let G be an additive group. Computational Diffie-Hellman Problem (CDHP) in G is defined by:

Definition 2.9. Given P , aP and $bP \in G$ for $a, b \in \mathbb{Z}_p^*$, compute abP .

Decisional Diffie-Hellman Problem (DDHP) in G is defined by:

Definition 2.10. Given P , aP , bP and $cP \in G$ for a, b and $c \in \mathbb{Z}_p^*$, decide whether $c \equiv ab \pmod{p}$ or not. If true, (P, aP, bP, cP) is called a Diffie-

Hellman tuple.

A group G with prime order is called Gap-Diffie-Hellman (GDH) group if DDHP can be solved in polynomial time while no known polynomial time algorithm can solve CDHP with non-negligible advantage in polynomial time. An additive group G constructed on an elliptic curve over finite fields is such a GDH group due to the existence of the bilinear pairings. MOV attack [125] based on Weil pairing and FR attack [86] based on Tate pairing make it possible to solve DDH in G , whereas there is no known polynomial time algorithms for CDHP in G . CDHP and DDHP are the fundamentals of pairing-based cryptosystems. The security of many cryptosystems from the bilinear pairings are derived from them (see [79]).

2.7 Summary

This chapter presents an introduction of the mathematical background for the modern cryptography. Firstly, the definition of provable secure signature scheme is described and the related security models are introduced. Secondly, the basic mathematical knowledge of the modern cryptography, such as the definitions of Group and Finite Field, are introduced. Thirdly, two cryptographic assumptions, RSA problem and ECDLP, are introduced. Finally, some preliminaries of pairing-based cryptography derived from ECC are introduced.

Chapter 3

A Practical Group Signature Scheme From RSA

3.1 Introduction

The concept of group signatures was first introduced by Chaum and van Heyst [61]. In such signatures, participants are classified as either group manager or group members. Group manager plays the authority role and does not sign messages, while group members are message signers. A group member can generate a signature on a message on behalf of the group using its own group member secret key and without revealing its identity. Those who have the corresponding group public key can check the validity of a group signature, but will not know the identity of the signer. This property is known as the anonymity of group signatures. If a dispute occurs, the group manager can open the disputed signature and extract the identity of the signer from this disputed signature by using a group secret key. This is another basic property of group signatures:

traceability. Usually, a group consists of one group manager and a number of group members. Besides the basic properties of anonymity and traceability, other useful properties that enable the use of group signature schemes for real applications include unlikability, unforgeability, collusion resistance, exculpability and frame resistance. These properties will be introduced in Section 3.2.

Group signature schemes can be classified as two types: *static* group signature schemes and *dynamic* group signature schemes. In a static group signature scheme, the number of group members and their identities is fixed when the group is established. No new group member is allowed to join the group. A dynamic group signature scheme, however, allows new users to join the group or allows a group member to leave the group in real-time. If a group supports either joining or leaving, it is called a *partially* dynamic group. If a group supports both join and leave, it is called a *fully* dynamic group. An efficient fully dynamic group signature scheme should satisfy the following requirements: a) The public key size and the signature length should be constant even if the number of group members grows; b) Signing a message and verifying a signature should be as fast as possible; c) The group signature size should be as short as possible; d) Joining and leaving a group should be as fast as possible. Recently, many applications which require the properties of group signature schemes have appeared. These include Trusted Computing [165], privacy preserving attestation [90] and the Vehicle Safety Communications System (VSCS) [112].

In early group signature schemes based on RSA or DLP [61, 63, 50], both the size of the group public keys and the group signature's length increased with the growth of the group members. Although some of these schemes were secure,

they were impractical. Other group signature schemes [51, 47, 48, 49, 8, 74, 137] provided constant-size group public key and group signatures. However, they were also not practical because of inefficiencies in signing messages and verifying signatures, or because of their complicated procedures for joining and leaving the group. Furthermore, some special applications required that the group signature schemes had to provide short group signatures. For example, VSCS requires that the group signature length must be not more than 250 bytes. Although the concept of group signatures was proposed for some years, only a few schemes can generate short group signatures. Boneh, Boyen and Shacham [34] successfully reduced the length of group signatures to 1533 bits (or 192 bytes) at the 1024 bits security level by using a short group signature schemes based on the bilinear pairings. However, their scheme requires point compression, which means that extra cost had to be paid in the form of uncompressing these points when verifying group signatures. Until now, the fastest group signature schemes were constructed by [10, 6, 46]. However, they are not suitable for VSCS due to extremely long group signature length. Thus, the need to construct a practical and efficient group signature scheme with a shorter signature length is an interesting issue. This chapter will address this issue.

3.1.1 Contribution

We introduce a new approach to construct a practical group signature scheme with short signatures in the random oracle model. Our idea is built on the idea of double authorities by Bellare et al. [26] and using a trusted online third party in RSA signature by Boneh et al. [35]. As the second authority, a new participant - the trusted online third party called the *security mediator* [35, 73] -

is introduced into group signatures. By cooperating with the original authority, the group manager, the *security mediator* makes it easy to implement an efficient group signature scheme with short signatures. The group signature scheme with the security mediator also simplifies the procedure of signing messages, verifying signatures, concurrently joining a group, opening group signatures and immediately revoking group membership. An informal security analysis described in Section 3.3.3 shows that the proposed group signature scheme meets all the requirements of a group signature scheme. The proposed practical group signature scheme is constructed based on the classical RSA problem. Similar to other efficient group signature schemes, it also provides a constant size group public key and signatures of constant length even if the size of the group is increased.

3.1.2 Prior Research

Following the initial publication by Chaum and van Heyst [61] where four group signature schemes were described, many group signature schemes, such as [63, 50, 51, 47, 48, 49, 10, 8, 74, 19, 34, 137, 26], have been proposed. Early group signature schemes [61, 63, 50] possessed the basic properties of group signatures, such as computational anonymity, information theoretical anonymity, and adding/revoking group members. However, these schemes have some shortcomings in terms of performance. For example, all the four schemes described in [61] cannot distribute the functionality of the group manager efficiently. In the two schemes described in [63], the group manager is allowed to misattribute a signature such that a group member who has not signed a message may be falsely accused. In addition, in the schemes [61, 63, 50], the size of the group public keys

and the length of the group signatures increase with the growth of the number of the group members. The authors of [51, 47, 48] addressed these shortcomings and proposed excellent solutions such that their new group schemes provided constantly size group public key and constant signature lengths. Since then, new group signature schemes [10, 8, 34, 74, 137] have constant key sizes and fixed signature lengths. The group signature scheme proposed by Ateniese and Tsudik [7] not only corrected the insecure group signature scheme proposed by [48], but also developed a practical multi-group signature and a sub-group signature. In [7], the authors provided a solution for re-issuing group and individual members' keys. In [10], Ateniese et al. introduced a new secure group signature scheme to protect against coalition attacks, which many previous group signature schemes such as [63, 50, 51] were susceptible to. Ateniese and Medeiros [8] introduced the first group signature scheme with constant-size parameters where no group member, including the group managers, needed to know trap-door secrets. More importantly, their scheme works in various groups where the Decision Diffie-Hellman assumption holds (see Chapter 2).

The group signature schemes introduced in [34] and [137] both use bilinear pairings. The latter provided two secure and efficient group schemes which also generated shorter group signatures than [10], while the former provided an efficient scheme with shorter group signatures than the latter. The theoretical foundation for the static group signature scheme was built by Bellare et al. [19], and Bellare et al. [26] soon established the theoretical foundations for the dynamic group signature scheme. In particular, [26] first constructed group signature schemes using two authorities so as to provide more security in face of the possibility that authorities may be dishonest.

The concept of the security mediator was first introduced for mRSA (Mediated RSA) by Boneh et al. [35] where a user's RSA private (secret) key is split into two parts: one part is assigned to the user, the other is assigned to the security mediator. Neither the user and the security mediator can sign or decrypt messages without the other's help because neither knows the factorization of the RSA modulus. Based mRSA, Ding and Tsudik [73] proposed the idea of IB-mRSA (Identity-based Cryptography with mRSA) which blends the features of identity-based and mediated cryptography. Their security proof showed that IB-mRSA with OAEP (Optimal Asymmetric Encryption Padding) [22] in the random oracle model is as secure as standard RSA with OAEP in the random oracle model.

The rest of this chapter is organized as follows. In Section 3.2, we introduce the definition and the security notion of a group signature scheme. In Section 3.3, a new practical group signature scheme with security analysis is given. In the same section, we also discuss the signature length of the proposed group signature scheme. In Section 3.4, we summarize the new practical group signature scheme.

3.2 Definition, Security Requirements and Security Mediator

In this section, we first introduce the basic definition of a group signature. We then depict the requirements of a secure group signature scheme. In the end of this section, the concept of the security mediator introduced by Boneh et al. [35] is reviewed.

3.2.1 Definition

In a traditional dynamic group signature scheme, participants are of two types: one participant is a designated *group manager* who play the role of an authority, while the others are *group members* who play the role of the signers. Usually, there is only one group manager while the number of group members is not restricted. A group signature scheme is composed of five algorithms **GSetup**, **GJoin**, **GSign**, **GVerify**, **GOpen** which are defined as follows:

- **GSetup**: This is a probabilistic algorithm that takes as input a security parameter $\kappa \in \mathbb{N}$ and an integer l (where l denotes the number of the members in this group). It outputs the system parameters, a group public key, a corresponding group secret key and l membership secret keys. Each group member obtains a membership secret key and keeps it secret.
- **GJoin**: This is a protocol between the group manager and a user who want to join the group. It is used to decide whether the group accepts this user as a member or rejects him. If this user successfully joins the group and becomes a member, then he will immediately obtains its membership certificate and the group membership secret key from the group manager.
- **GSign**: This is a probabilistic algorithm that takes as its input a group public key, a membership certificate, a secret key of a member of the group and a message. Its output is a group signature based on the given message.
- **GVerify**: This is a deterministic algorithm that takes as its input a group public key, a message and a candidate signature, and outputs a boolean

value. If the candidate signature is valid, this algorithm returns 1 (or true). Otherwise, it returns 0 (or false).

- **GOpen**: This is a deterministic algorithm that takes as its input a message, a valid group signature and the group secret key. Its output is the identity of the signer. This algorithm is run only by the group manager.

3.2.2 Security Requirements

Early work [61, 51, 7] introduced certain security requirements for a secure group signature scheme. However, people always found some new methods to these schemes. Bellare et al. [26] gave a security model for the dynamic group signature scheme with two authorities in which the group signature scheme is secure only if it satisfies the three security requirements: anonymity, traceability and non-frameability. There are two authorities in the proposed group signature scheme, but the function of each authority in the proposed scheme is different with that in the scheme [26]. It is difficult to apply the security model of Bellare et al. [26] directly to prove the security of the proposed scheme. We have to choose another security model. Ateniese et al. [10] provided a model of a *secure* group signature scheme which can resist all existing attacks such as coalition attack [47] (but the security of the proposed group signature scheme requires an assumption which will be introduced in the next section). This model required that a secure group signature scheme should satisfy the following security properties:

- **Correctness**: If a group signature generated by a group member using **GSign** is valid, then it must be accepted by **GVerify**.

- **Unforgeability:** No one except group members can generate valid group signatures on behalf of the group. Any user outside the group cannot generate valid signatures on behalf of the group.
- **Anonymity:** Given a valid group signature on some message, it is computationally infeasible for anyone except the group manager to identify the original signer.
- **Unlinkability:** Given several group signatures on the same or different messages, it is computationally infeasible for anyone except the group manager to decide whether the signatures were generated by the same or different group members.
- **Exculpability:** A signature generated by a group member cannot be successfully attributed to others. In other words, neither the group manager nor group members can generate signatures on behalf of other group members.
- **Traceability:** The group manager is always able to open a valid group signature and find out who actually signed the message. No signer is allowed to prevent the group manager from opening a valid signature.
- **Coalition-resistance:** Even if a coalition of some group members (or the entire group) collaborates to generate a valid group signature on some message, the group manager can still attribute this signature to the colluding members.

3.2.3 Security Mediator and mRSA

In mRSA proposed by Boneh et al. [35], the security mediator is defined as follows:

Definition 3.1. *A security mediator is an online semi-trusted server which can serves many users. To sign or decrypt a message, the user must first obtain a message-specific token from the security mediator. To revoke the user's ability to sign or decrypt, the security mediator must stop issuing tokens for the user's public key.*

For the proposed group signature scheme, an important assumption must be made: any adversary is unable to comprise a security mediator or the security mediator cannot be comprised by any adversary. Only depending on this assumption, we can prove the security of the proposed group signature scheme under the security model introduced in the previous section. Without this assumption, the proposed group signature scheme would offer no security. If a security mediator is comprised, all the users' keys will leak out and the adversary obtains easily all the users' private key.

mRSA is constructed based on the Security Mediator. Similar with the general RSA, mRSA is used to encrypt messages and generate signatures. mRSA signature is composed of three algorithms: **mRSA Key Setup**, **mRSA Sign** and **mRSA Verify**. Let SM denote the security mediator, mRSA signature scheme is described as follows:

- **mRSA Key Setup:** A trusted third party such as **CA** is in charge of generating all key setup. **CA** generates the modulus n_i from two large prime numbers (p_i, q_i) and a public/secret key pair (d_i, e_i) for each user in the

same manner as in standard RSA. Then **CA** picks a random value d_{S_i} in the interval $[1, n_i]$ and computes $d_{U_i} = d_i - d_{S_i} \pmod{\phi(n_i)}$ where $\phi(n_i) = (p_i - 1)(q_i - 1)$. d_{S_i} is secretly communicated to a SM and d_{U_i} is secretly communicated to the user U_i , (n_i, e_i) is published as the public key.

- **mRSA Sign:** The user U_i first computes M by hashing the message m , then sends M to the SM. SM checks whether the user U_i is a revoked user or not. If not, SM computes a partial signature $\sigma_{S_i} = M^{d_{S_i}} \pmod{n_i}$ and replies with it to U_i . U_i computes $\sigma_{U_i} = M^{d_{U_i}} \pmod{n_i}$ and $\sigma_i = \sigma_{U_i} \cdot \sigma_{S_i} \pmod{n_i}$. The signature on the message m is set to σ_i . If $M = \sigma^{e_i} \pmod{n_i}$.
- **mRSA Verify:** For a given message-signature pair (m, σ) , the verifier computes the hashed value M from the message m and $M' = \sigma^{e_i} \pmod{n_i}$. The signature is valid if $M = M'$.

3.3 A New Group Signature With Security Mediator

Bellare et al. [26] first used two authorities, an issuer and an opener, to construct full dynamic group signature schemes. In their model, the authorities are endowed with different functions and each has its own secret key. The opener can open signatures while the issuer interacts with a user to issue him a signing key. Motivated by their idea, we propose a group signature scheme with two authorities. Besides keeping the group manager as an authority, we introduce a trusted online third party, called a *security mediator* (SM) into the group

signature. This trusted online third party plays the role of the second authority in the group signature scheme.

Similar to mRSA which splits a user's RSA private key between the user and the security mediator, the proposed group signature scheme splits the group secret key d , originally kept by the group manager, into two parts for every group member: one part d_{U_i} is given to the group member U_i as its group membership secret key, and the other part d_{S_i} is sent to SM and kept secret by him. Neither the group member nor SM can sign a message solely. In other words, SM cannot sign a message without the permission of some group member and a group member cannot sign a message without the permission of SM. In the proposed group signature scheme, a group member must cooperate with SM to generate a valid group signature and cannot generate a valid group signature by himself. The group manager plays the role of "the issuer" defined in [26]. SM plays the role of "the opener" but is endowed with more duties. He not only opens the disputed signature but also interacts with group members to generate signatures. He offers the following functionalities:

- Help the group manager and the users who want to become group members realize the "join" protocol.
- Cooperate with the valid group members to generate valid group signatures.
- Help the group manager revoke the membership of a group member.
- Open some group signatures and reveal the identities of the original signers when a dispute happens.

The appearance of SM makes it simple to join a group or revoke a group

member from the group. To join a group, the group manager splits the group secret key d into two parts d_{U_i} and d_{S_i} after receiving a request from a user who wants to join the group, sends the two parts to the user and SM, respectively. After the user obtains its own secret key d_{U_i} and SM obtains the corresponding key d_{S_i} , the user becomes immediately a member of the group and is able to sign messages anonymously on behalf of the group. This procedure is very simple and does not require other parameters, such as the group public key and other members' group membership secret keys, to be modified. To revoke the membership of a group member U_i , the group manager only needs to inform SM to delete the corresponding d_{S_i} such that SM cannot provide the group member partial signatures. The group membership can therefore be revoked immediately even if this user still owns a group membership secret key.

In the next section, we will illustrate how to construct a simple and practical group signature scheme with the help of SM.

3.3.1 The Proposed Group Signature Scheme

The new group signature scheme is proposed in this subsection. This new scheme introduces the Security Mediator (SM) architecture into the group signature. This idea is from the double authorities proposed by Bellare et al. [26] and Mediated RSA by Boneh et al. [35] and Ding et al. [73]. Involved in this new scheme are the group manager for initializing key generation, another authority called the security mediator for assisting users to sign messages and a body of users. Every user can generate signatures on behalf of the group. The group signature scheme are specified by five algorithms: **GSetup**, **GJoin**, **GSign**, **GVerify** and **GOpen**. The following is the details of the new scheme:

- GSetup:** Given a security parameter $\kappa \in \mathbb{N}$, the group manager first picks a κ bits RSA modulus n_1 which is the product of two large prime numbers p_1 and q_1 , and computes $\phi(n_1) = (p_1 - 1)(q_1 - 1)$. Then, he picks a random integer $e_1 < \phi(n_1)$ such that $\gcd(e_1, \phi(n_1)) = 1$ and computes the integer d_1 such that $d_1 \cdot e_1 = 1 \pmod{\phi(n_1)}$. Simultaneously, SM picks another κ bits RSA modulus n_2 which is the product of two large prime numbers p_2 and q_2 . It is noted that $n_2 < n_1$. Next, SM computes $\phi(n_2) = (p_2 - 1)(q_2 - 1)$. SM then picks a random integer $e_2 < \phi(n_2)$ such that $\gcd(e_2, \phi(n_2)) = 1$ and computes the integer d_2 such that $d_2 \cdot e_2 = 1 \pmod{\phi(n_2)}$. d_1 and d_2 are kept secret by the group manager and SM, respectively. The group public key is a tuple $(e_1, e_2, n_1, n_2, H_1, H_2)$, where $H_1: \{0, 1\}^*$ is a standard collision-resistant hash function such as SHA-224 [136] or RIPEMD-256 [75], and $H_2: \{0, 1\}^* \rightarrow \mathbb{Z}_\kappa$ is a hash function whose output is a κ -bit number.
- GJoin:** Let U_i be a user who wants to become a member of a group. He sends a request to the group manager of the group that he want to join. Suppose SM and the group manager are trusted in this group. The group manager chooses a random number d_{U_i} from $\mathbb{Z}_{n_1}^*$ and computes $d_{S_i} = d_1 - d_{U_i} \pmod{\phi(n_1)}$. d_{U_i} is sent to U_i as the group membership secret key and (d_{S_i}, U_i) is sent to SM and kept secret. After the user and SM obtain their corresponding secret key, U_i becomes a group member with a group membership secret key d_{U_i} . It is noted to protect against the potential attacks (the details are described in the next section), when the group manager allocates the membership secret keys for those who want to join the group, the following rules must be required:

- $d_{U_i} \neq d_{U_j}$ when $i \neq j$.
 - $d_{U_1} + d_{U_2} + \cdots + d_{U_j} \neq d_1 \pmod{\phi(n_1)}$ for any positive integer j .
 - $d_{U_1} + d_{U_2} + \cdots + d_{U_j} \neq d_{U_l} \pmod{\phi(n_1)}$ for any positive integers j and l .
- **GSign:** To generate a group signature on a message $m \in \{0, 1\}^*$, the group member U_i must cooperate with SM to complete a signature:
 - U_i first picks a random element λ from $\mathbb{Z}_{n_1}$ and computes $M = H_2(\lambda || m)$. Then he computes $h'_i = M^{d_{U_i}} \pmod{n_1}$, then sends M , h'_i along with its identity U_i to SM.
 - SM first sends a request to the group manager to check whether U_i 's membership is available or not. If U_i 's membership is approved by the group manager, SM searches its storage list and finds the corresponding d_{S_i} , then computes $r'_i = M^{d_{S_i}} \pmod{n_1}$ and checks whether the equation $(r'_i \cdot h'_i)^{e_1} = M$ holds or not. If so, SM stores (U_i, h'_i, d_{S_i}) and computes $S_i = h_i'^{d_2} \pmod{n_2}$ and $\sigma_{S_i} = S_i^{d_{S_i}} \pmod{n_1}$. (S_i, σ_{S_i}) is sent back to U_i .
 - After receiving (S_i, σ_{S_i}) , U_i first checks whether the equation, $h'_i = S_i^{e_2} \pmod{n_2}$, holds or not. If not, U_i stops. Otherwise, U_i computes $\sigma_{U_i} = S_i^{d_{U_i}} \pmod{n_1}$, $h_i = H_1(m || h'_i)$ and $\sigma = \sigma_{U_i} \cdot \sigma_{S_i} \pmod{n_1}$. Finally, the group signature on the message m is (h_i, σ) . The details of GJoin and GSign are shown in Fig. 3.1.
 - **GVerify:** Given a group signature pair (h, σ) on a message m , the verifier computes $S = \sigma^{e_1} \pmod{n_1}$ and $h'' = S^{e_2} \pmod{n_2}$. The group

signature is accepted if the following equation holds

$$h = H_1(m||h'').$$

- **GOpen**: When some dispute happens, SM has to open a group signature on a message to trace the identity of the signer. Suppose that SM wants to open a group signature pair (h, σ) on some message m . He first computes $S = \sigma^{e_1} \pmod{n_1}$ and $h' = S^{e_2} \pmod{n_2}$ to verify the validity of the group signature. If the disputed group signature is valid, SM searches the storage list by the index h' to find the corresponding (U_i, d_{S_i}) and reveal the identity of the actual signer.

Performance Signing a message, verifying a signature, or opening a signature only needs simple multiplications on $\mathbb{Z}_{n_1}^*$ and basic modulo exponentiations. The implementation of the proposed group signature scheme therefore is very simple. To join a group, even modulo exponentiation is not needed and only a simple addition is required. In the **GSign** phase, seven modulo exponentiations and four multiplications are needed. In the **GVerify** phase, two modulo exponentiations and one multiplication are needed. In the **GOpen** phase, two modular exponentiations and one multiplication are needed.

Remark 3.1. *We note that the scheme is completely insecure when two or more users share a same modulus in classical RSA signature scheme since the knowledge of a single secret-public key pair allows to factor the modulus. However, the proposed group signature scheme allows multiple group members to share the same modulus since neither the group member nor SM is able to know the*

Figure 3.1: GJoin and GSign in the proposed group signature scheme

^aGM: the group manager^bSM: the security mediator

complete group secret key corresponding to the group public key. A collusion of the modulus between a group member and SM would result in the crash of the proposed group signature scheme. Therefore, SM here must be assumed to be totally trusted and no user is able to compromise it.

3.3.2 Revocation

As described above, after receiving a request for a group signature from a user U_i , SM first sends a query to the group manager to check the validity of the membership of the user U_i . Once the membership of U_i is confirmed by the group member, SM performs the next operation and completes the group signature together with U_i . The group manager maintains a list called *GL-list* where $\langle U_i, d_{U_i}, d_{S_i} \rangle$ are stored for all the legal members. SM also maintains a list called *SL-list* where $\langle U_i, d_{S_i} \rangle$ are stored for all the legal members. When the group manager decides to revoke some member U from the group, the name of U is immediately removed from *GL-list* and SM is informed to delete the corresponding $\langle U, d_{S_i} \rangle$ from *SL-list*. When a revoked user U requests SM to sign a message, SM will deny the request of U because U 's membership cannot be approved by the group manager. Thus the proposed group signature scheme provides an efficient method for revoking group members. The proposed group signature scheme not only allows group members to leave the group, but also allows users who are not group members to join the group. The number of the group members and their identities do not need to be fixed in the *GSetup* phase any more. It can be seen as a full dynamic group signature scheme.

Another advantage of the proposed group signature scheme is that both joining and leaving the group are carried out in real-time. Any entity who want

to join the group can obtain a secret signing key at once, and the group manager can revoke a legal member immediately.

3.3.3 Security Analysis

This section describes how the proposed group signature scheme satisfies all the requirements of a secure group signature scheme. **Correctness:** Suppose that

(h, σ) is a group signature pair on some message m generated by a group member U_i using **GSign**. σ is seen as the combination of three variables: a common variable $C = h'^{d_2} \pmod{n_2}$ computed by SM where $h' = (H_2(\lambda || m))^{d_{U_i}} \pmod{n_2}$ for a random $\lambda \in \mathbb{Z}_{n_1}$, a partial variable $\sigma_{S_i} = C^{d_{S_i}} \pmod{n_1}$ generated by SM and a partial variable $\sigma_{U_i} = C^{d_{U_i}} \pmod{n_1}$ generated by U_i . The following formula shows how to generate σ :

$$\begin{aligned} \sigma &= \sigma_{S_i} \cdot \sigma_{U_i} \pmod{n_1} \\ &= C^{d_{S_i}} \cdot C^{d_{U_i}} \pmod{n_1} \\ &= C^{d_1} \pmod{n_1} \end{aligned}$$

The verifier uses **GVerify** to computer S and h'' as follows:

$$S = \sigma^{e_1} \pmod{n_1} = C^{d_1 \cdot e_1} \pmod{n_1} = C$$

and

$$\begin{aligned}
 h'' &= S^{e_2} \pmod{n_2} \\
 &= C^{e_2} \pmod{n_2} \\
 &= h'^{(d_2 \cdot e_2)} \pmod{n_2} \\
 &= h'
 \end{aligned}$$

Finally, the verifier justifies the validness of a received signature by checking the correctness of the following equation:

$$h = H_1(m || h') = H_1(m || h'')$$

If the above equation is correct, it is concluded that a valid group signature generated by **GSign** can be accepted by **GVerify**. Otherwise, the received signature is invalid and the verifier discards it. Therefore, the verifier can use **GSign** to justify whether the signatures generated by **GSign** is valid. The proposed group signature scheme meets the requirement of the correctness property.

Unforgeability: To prove the unforgeability of the proposed group signature scheme, we first review the security of mRSA signature described in Boneh et. al [35] (Section 4). In mRSA signature, the attacker's goal is to forge the user's signature on the message m . A token, $\sigma_{S_i} = M^{d_{S_i}} \pmod{n_i}$, is sent back to the user from SM. The attacker can see M and the token σ_{S_i} . Boneh et al. [35] made a claim: “the knowledge of M and σ_{S_i} is of no use to an attacker, because d_{S_i} is just a random number in $[1, n_i]$ and it is independent of the rest of the attacker's view”¹.

¹Section 4 of [35] shows that this statement can be proven using a simulation argument.

Compared **GSign** of the proposed group signature scheme with **mRSA Sign** described in Section 3.2.3, **GSign** can be seen as a “two-round” **mRSA Sign**. In the first round, U_i plays the role of SM in **mRSA Sign** and computes $h' = M^{d_{U_i}} \pmod{n_1}$ under its secret key d_{U_i} , then sends it to SM. SM plays the roles of U_i of **mRSA Sign**. It first validates the received message h'_i from U_i under its own secret key d_{S_i} . Only when h'_i is validated, SM goes to the next step. In the second round, SM computes the partial signature σ_{S_i} on “the hashed message” S_i and sends (S_i, σ_{S_i}) to U_i . After receiving σ_{S_i} , U_i first validates it, and then computes σ_{U_i} on “the hashed message” S_i . Finally, I_i use the hash value h' and the product σ of σ_{S_i} and σ_{U_i} to generate the group signature. The forger’s goal is to forge (h', σ) on a message m . In both rounds, the forger can obtain the knowledge of (M, h') and (S_i, σ_{S_i}) . Based on Boneh et al’s claim, (M, h') is of no use to obtaining d_{U_i} , and (S_i, σ_{S_i}) is of no use to obtaining d_{S_i} . Without the knowledge of d_{U_i} and d_{S_i} , any forger cannot generate forged signatures. Even if the forger intercepts (M, h') from U_i (or (S_i, σ_{S_i}) from SM) and sends a forged pair $(\underline{M}, \underline{h'})$ to SM (or a forged pair $(\underline{S_i}, \underline{\sigma_{S_i}})$ to U_i , SM (or U_i) can easily find them by the validation and discard them. Therefore, the proposed group signature scheme satisfies the unforgeability property.

Anonymity: Suppose that the pair (h, σ) is a valid group signature on a message m generated by an unknown user U . The description below shows that no one can obtain the identity of the signer from h and σ . The first part h is obtained by $H_1(m||h')$ where $h' = (H_2(\lambda||m))^{d_U} \pmod{n_1}$. Without any loss of generality, we can even assume that h' is public. h' is a RSA signature under a secret key d_U . However, there does not exist a corresponding public key e_U for the group membership secret key d_U such that $e_U \cdot d_U = 1 \pmod{n_1}$. Thus,

no one can obtain the identity of the signer from h and h' .

The second part σ is the product of two elements in $\mathbb{Z}_{n_1}$. It is always a signature on “the message” S where $S = h'^{d_2} \pmod{n_2}$. To show the anonymity of the proposed group signature scheme, we consider the case where two different users U_i and U_j sign the same message. A group signature on the message m generated by group member U_i is:

$$\begin{aligned} h_i &= H_1(m || h'_i) \text{ where } h'_i = (H_2(\lambda_i || m))^{d_{U_i}} \pmod{n_1} \\ \sigma_i &= \sigma_{S_i} \cdot \sigma_{U_i} \pmod{n_1} \\ &= (h_i'^{d_2})^{d_{S_i}} \cdot (h_i'^{d_2})^{d_{U_i}} \pmod{n_1} \\ &= h_i'^{d_2 \cdot d_1} \pmod{n_1}. \end{aligned}$$

The group signature on the same message m generated by another group member U_j is:

$$\begin{aligned} h_j &= H_1(m || h'_j) \text{ where } h'_j = (H_2(\lambda_j || m))^{d_{U_j}} \pmod{n_1} \\ \sigma_j &= \sigma_{S_j} \cdot \sigma_{U_j} \pmod{n_1} \\ &= (h_j'^{d_2})^{d_{S_j}} \cdot (h_j'^{d_2})^{d_{U_j}} \pmod{n_1} \\ &= h_j'^{d_2 \cdot d_1} \pmod{n_1}. \end{aligned}$$

Obviously, σ_i and σ_j are decided by h'_i and h'_j , respectively. As in the previous discussion, no one can obtain any information about the identities of the signers from h'_i or h'_j . Therefore, a group signature pair (h, σ) does not include any information on the original signer. In other words, it is computationally hard for anyone, except the group manager, to discover the identity of the actual signer. Thus, the proposed group signature scheme satisfies the requirement of

anonymity.

Unlinkability: As shown above in the discussion on the anonymity property, no one except the group manager can uncover information on the signer from a given valid group signature pair, as this pair does not contain any information on the original signer. Therefore, given several valid group signatures, it is difficult to determine whether they were generated by the same group member. Therefore, the proposed group signature scheme satisfies the requirement of the property of unlinkability.

Exculpability: With the introduction of SM into group signatures, the requirement of this property is different with what was described in Section 3.2. Here, the requirement on SM, as well as a group member or the group manager, is that SM cannot solely generate a valid signature on behalf of other group members. Firstly, we show that a group member cannot sign messages on behalf of other group members. When a group member U_i has signed a message m and generated a valid signature pair (h_i, σ) , its identity is sent to SM by a trusted channel and is stored in the SL -list by SM. Because d_{U_i} is not equivalent to d_{U_j} when $i \neq j$, U_j cannot generate a valid group signature under U_i 's secret key d_{U_i} . If U_j sends $h' = M^{d_{U_j}}$ with the identity U_i , SM cannot validate h' under the secret key d_{s_i} and the message (h', U_i, M) will be discarded. Therefore, no group member can sign messages on behalf of other group members. Secondly, we show that even the group manager cannot sign messages on behalf of a group member. The group manager has the knowledge of the user's secret key d_{U_i} of U_i , SM's group secret key d_{S_i} for U_i and the group secret key d_1 . However, he does not know the secret key d_2 of SM. The second part σ of the group signature is an RSA signature on "the message" S using the group secret key d_1 . S

is a RSA signature on the message h' under SM's secret key d_2 . If the group manager can generate a valid signature on behalf of a group member, then he must be able to break a RSA signature scheme. Since this is impossible, a group manager is unable to sign messages on behalf of a group member. Thirdly, we show that SM cannot sign messages on behalf of other group members. SM has the knowledge of U_i , d_2 , d_{S_i} , h_i , h'_i and S_i (where $S_i = h_i^{d_2} \pmod{n_2}$). To generate a valid signature on behalf of U_i , SM must find a $\sigma' = S_i^{d_{U_i}} \pmod{n_1}$ such that $\sigma' \cdot S_i^{d_{S_i}} = S_i^{d_1} \pmod{n_1}$. However, SM does not know d_{U_i} and d_1 . σ' and $S_i^{d_1}$ are two RSA signatures created by using the secret keys d_{U_i} and d_1 , respectively. If SM is able to sign messages on behalf of U_i , that would indicate that he is able to break an RSA signature scheme. From the above discussion, the proposed group signature scheme satisfies the requirement of exculpability.

It is noted that the group manager and SM can sign messages on behalf of other group members when they cooperate. This case should be prohibited.

Traceability: All the group signatures can be generated only with the help of SM, and SM has stored the identities of the original signers at the time during signing messages. When a dispute occurs, SM can open the group signature and identify the actual signer as mentioned in the description of **GOpen** in Section 3. From the assumption that SM cannot be comprised by any adversary, only SM is always able to open a valid group signature and identify the actual signer. Thus, the proposed group signature scheme satisfies the traceability requirement.

Coalition-resistance: We consider two possible cases where some group members could collaborate to forge group signatures.

- **Case 1:** Suppose the secret key of the user U_i is denoted by d_{U_i} . If $d_{U_1} +$

$d_{U_2} + \dots + d_{U_j} = d_1 \pmod{\phi(n_1)}$ for some positive integer j , then the set of users $\mathbb{U} = (U_1, U_2, \dots, U_j)$ can collaborate to generate valid group signatures and the group manager will not be able to identify the original signers. Suppose that one member in $\mathbb{U}$ imitates the group manager and sends SM an SM secret key $d_{S'_i}$ for a user U'_i who does not actually exist. This member would then be able to compute $d_{U'_i} = d_1 - d_{S'_i}$ and sign messages under $d_{U'_i}$. SM will not be able to discern that this request for signing from U'_i is illegal. He thus cooperates with the user set $\mathbb{U}$ and generates a valid group signature which can be accepted by the verifier. However, the group manager will be unable to identify this signer.

- **Case 2:** If $d_{U_1} + d_{U_2} + \dots + d_{U_j} = d_{U_l} \pmod{\phi(n_1)}$ for some integers j and l , then user set $\mathbb{U} = (U_1, U_2, \dots, U_j)$ could collaborate to generate a valid group signature on behalf of U_l . Let U_t ($1 \leq t \leq j$) be a member of $\mathbb{U}$. Given a message $\tilde{m}$, U_t picks a random element $\tilde{\lambda}$ from $\mathbb{Z}_{n_1}$ and computes $\tilde{M} = H_2(\tilde{\lambda}||\tilde{m})$. Then all members of $\mathbb{U}$ compute $\tilde{h}'_i = \tilde{M}^{\sum_{i=0}^j d_{U_i}} \pmod{n_1}$ and $\tilde{h}_i = H_1(\tilde{m}||\tilde{h}'_i)$. U_t sends $(\tilde{h}_i, \tilde{h}'_i, \tilde{M})$ with the identity U_l to SM. After SM confirms the validity of U_l from the group member, he computes

$$\tilde{S} = M^{d_2} \pmod{n_2} \text{ and } \tilde{\sigma}_{S_l} = \tilde{S}^{d_{S_l}} \pmod{n_1}$$

Each group member U_i computes $\tilde{\sigma}_{U_i} = \tilde{S}^{d_{U_i}} \pmod{n_1}$. Let

$$\tilde{\sigma}_{U_l} = \prod_{i=1}^j \tilde{\sigma}_{U_i} = \prod_{i=1}^j \tilde{S}^{d_{U_i}} \pmod{n_1} = \tilde{S}^{d_{U_l}} \pmod{n_1}$$

and

$$\tilde{\sigma}_l = \tilde{\sigma}_{U_l} \cdot \tilde{\sigma}_{S_l}^s \pmod{n_1}$$

It is obvious that $(\tilde{h}_l, \tilde{\sigma}_l)$ is a valid group signature. The group members $U_1, U_2, \dots, U_j$ have collaborated to generate a valid group signature on behalf of U_l . However, in the proposed group signature scheme, the above cases are not permitted. Therefore, no group members can generate valid group signatures without the cooperation of SM. Even if some group members (including a whole set of the entire group, together with the group manager) collaborate, they will be unable to generate a valid group signature. The proposed group signature scheme satisfies the requirement of the property of coalition-resistance, since the two cases above have been avoided.

3.3.4 Signature Size

In Table 3.1, we compare the proposed group signature scheme with the short group signature scheme proposed by [34] (which is called BBS scheme for short) in terms of the group signature length at different security levels. The first column shows the security level, while the second and third columns show the size of the group signature of the proposed group signature schemes and BBS scheme, respectively.

Table 3.1: Comparison of the group signature length between different schemes

Security level (bits)	The proposed scheme (bits)	BBS scheme [34] (bits)
1024	1184/1248	1533
2048	2272	2019
3072	3328	2307

Suppose that BBS scheme is built on a non-supersingular elliptic curve of order p . The signature generated by BBS scheme is made up of three compressed

points and six elements in $\mathbb{Z}_p^*$. At the 1024 bits level, p is at least 170 bits and the embedding degree is 6 [34]. The length of signatures generated by BBS scheme is 1533 bits. The signature generated by the proposed group signature scheme is made up of a hash value and an element in $\mathbb{Z}_{n_1}$. We use SHA-224 [136] which provides 2048 bits security. The size of the signatures generated by the proposed group signature scheme is 1248 bits, which is shorter than the 1533 bits created by BBS scheme. The size of the group signature is reduced by approximately 19%. If we replace SHA-224 by RIPEMD-160 [75], the size of the signatures generated by the proposed group signature scheme is only 1184 bits. The size of the group signature is reduced by approximately 23%.

At the security levels of 2048 bits, the order p of elliptic curves is at least a 224-bits prime. The size of the signatures generated by the proposed group signature scheme is $(2048 + 224) = 2272$ bits. BBS scheme still requires three compressed points and six elements in $\mathbb{Z}_p^*$. The signature generated by BBS scheme is at least $(224 + 1) \times 3 + 224 \times 6 = 2019$ bits, which is shorter than the signature size of 2272 bits generated by the proposed group signature scheme. However, it is noted that the embedding degree k of elliptic curves is at least 10, such as to make $klg(p) \geq 2048$ for resisting index-calculus attacks [138]. The results in [109, 157] show that the growth of k will increase the time for computing the exponentiations (experimental results in Chapter 4 will confirm this). At the security levels of 3072 bits, to resist Pohlig-Hellman attack [138] and index-calculus attack [138], p is at least 256 bits and $klg(p)$ is at least 3072 bits. In this case, BBS scheme can generate the shortest signature only when $k = 12$ and the signature length is 2307 bits, which is shorter than the 3328 bits generated by the proposed group signature scheme. However, computing exponentiations will be very time consuming when $k = 12$. From the description

above, we can conclude that the proposed group signature scheme can be viewed as an appropriate alternative short group signature scheme when the security level is 1024 bits.

3.4 Summary

This chapter introduces a new participant, called the *security mediator*, into group signatures. Based on this new participant, we construct a practical group signature scheme from a classical RSA signature scheme in the random oracle model. The new scheme meets all the requirements of a secure group signature. The appearance of the *security mediator* makes it simple and efficient to join the group at any time, revoke membership immediately and open group signatures when dispute occurs. More important, the proposed group signature scheme is a short group signature scheme. In particular, group signatures generated by the proposed group signature scheme are shorter at the 1024-bits security level than those generated by other short group signature schemes such as Boneh et al. [34].

Chapter 4

Efficient and Secure Identity-Based Signature Schemes

In Chapter 3, we introduced a group signature scheme which is one of the traditional cryptographic schemes based on PKC. The public/secret key pair in the proposed group signature scheme may be generated randomly by a trusted third party CA. Recently, identity-based (ID-based) cryptosystems have made great progress. Compared with traditional PKC, the public key in identity-based cryptosystems is not random but fixed such as user identity and IP address. This simplifies the management of key distribution. In Chapter 4 and 5, we concentrate on designing new identity-based signature schemes from the bilinear pairings over elliptic curves. We also develop some applications utilizing these new identity-based signature schemes.

4.1 Introduction

In the traditional public key infrastructure (PKI), public keys are randomly generated by the key generation center or users. The identity of a user is bound with the public/secret key pair by a public key authentication framework such as the X.509 certificate framework [115]. Establishing, maintaining and managing this framework is not an easy task but a very complicated task. Shamir [161] proposed a method to avoid the use of such a public key authentication framework. He suggested that if public keys were designated as users identities, then the binding of identities to users and the corresponding public/secret key pairs would not be needed. This method, called ID-PKC, simplifies key management in comparison with traditional public key cryptography [143]. Although it has been more than twenty years since the concept of identity-based Public Key Cryptography (ID-PKC) was proposed by Shamir [161], and people can construct the simplistic “folklore” identity-based signature scheme by combining a conversational signature with a conversational certificate [93], it has been rarely discussed in the real-world applications because of the lack of efficient algorithms. However, the status changes due to the work of Boneh, Franklin [36] and Cocks [69].

Bilinear pairings used to be thought of as negative tools in Elliptic Curve Cryptography (ECC). Recently, Boneh and Franklin [36] constructed an efficient identity-based encryption (IBE) scheme using bilinear pairings over elliptic curves. Since then, research on ID-PKC from bilinear pairings has made great progress. Pairing-based cryptography has been one of the most active topics in the applied cryptography. Many identity-based cryptographic schemes have been published. Examples are identity-based encryption (IBE) schemes

[36, 37, 93], identity-based key agreement (IBKA) schemes [62, 163] and identity-based signature (IBS) schemes [153, 142, 54, 65, 100, 174, 175]. In particular, IBS has been suggested in the real world applications such as securing IPv6 neighbor and router discovery [4]. To satisfy the requirements of the real-world applications, researchers have designed new IBS schemes and improved their efficiency. Despite this, improving the efficiency of IBS scheme is still an interesting research issue. This is the first issue that we will address in this chapter.

The second issue is to build non-supersingular elliptic curves suitable for pairing-based cryptosystems. Most cryptographic schemes from bilinear pairings were constructed on supersingular elliptic curves when they were initially proposed. Examples include Joux's three-party key agreement protocol [103] and Boneh and Franklin's identity-based encryption scheme [36]. Supersingular elliptic curves are suitable for pairing-based cryptographic schemes because they provide the following benefits: a) The embedding degree on supersingular elliptic curves may be either 2, 3, 4 and 6, all of which conduce to computing pairing very efficiently. b) Supersingular elliptic curves offer an endomorphism, called distortion map [167] which makes it efficient to compute pairings. However, in the short signature scheme proposed by Boneh et al. [40, 41], the authors expressed some doubts about the security of supersingular elliptic curves: One is that the discrete logarithm problem in a finite field with a small characteristic is not as secure as a finite field with a large characteristic; the other is that the discrete logarithm problem on supersingular elliptic curves is susceptible to be attacked, although the authors do not provide concrete proof of this. To avoid possible attacks against supersingular elliptic curves and shorten the length of signatures, it is suggested that non-supersingular elliptic curves replace supersingular elliptic curves. Building non-supersingular elliptic curves suitable for

pairing-based cryptographic schemes is not an easy task as such curves have to satisfy several rigorous conditions. From the introduction of [11], only very few curves can satisfy these rigorous conditions. Given this, building suitable non-supersingular elliptic curves is one of the most interesting issues in elliptic curve cryptography¹.

4.1.1 Contributions

The first contribution is the proposal of a new IBS scheme in the random oracle model. In all existing cryptographic schemes using bilinear pairings, a special hash function called the *MapToPoint* function [40] is necessary. The *MapToPoint* function is used to map identity information (e.g. user name, IP address) or messages into points on elliptic curves. This special function is probabilistic and time consuming. In PKC 2004, Zhang et al. [182] modified BLS short signature [40] to obtain a fast short signature scheme without the *MapToPoint* function and called it ZSS scheme. The security of ZSS scheme depends on a mathematical assumption called k -CAA [129]. Motivated by their method, a new IBS scheme in the random oracle model which does not require *MapToPoint* function is first introduced. To prove the security of the proposed IBS scheme in the random oracle model, we extend k -CAA and present a new complexity assumption called Generalized k -CAA. The performance comparison shows that the proposed IBS scheme in the random oracle model provides faster key extraction and faster verification than the current IBS schemes.

The second contribution is to introduce a simple method to build non-

¹Because the first proposal for building suitable elliptic curves was proposed by Miyaji, Nakabayashi and Takano [130], non-supersingular elliptic curves suitable for pairing-based cryptosystems are also called MNT curves.

supersingular elliptic curves suitable for pairing-based cryptosystems. Two examples of non-supersingular elliptic curves are provided. This method depends on a new property of non-supersingular elliptic curves suitable for pairing-based cryptosystems which has not been described in prior literature. Furthermore, the proposed IBS schemes and other IBS schemes on supersingular or non-supersingular elliptic curves (including the new example curves) are implemented at different security levels. The experimental results show that the proposed IBS scheme in the random oracle model provides high performance at the 1024 bits and 2048 bits security levels.

4.1.2 Prior Research

A brief review of IBS schemes: All existing IBS schemes can be classified into two types: pairing-based IBS schemes, which are based on the bilinear pairings, and non-pairing-based IBS schemes, which are not based on the bilinear pairings. Most existing IBS schemes are pairing-based IBS schemes, with only a few being non-pairing-based IBS schemes. Some non-pairing-based IBS schemes lacked formal security proofs when they were presented. But in the later work [76, 20], researchers have provided enough support for their security. Below, we introduce these IBS schemes one by one.

The earliest three IBS schemes [161, 85, 98] were all non-pairing-based IBS schemes. In CRYPTO'84, Shamir [161] introduced the concept of ID-PKC and developed the first valid IBS scheme based on RSA, called Sh-IBS scheme. He explained why RSA could not be used to construct IBE. Sh-IBS scheme lacked a formal security proof until Bellare et al. [20] proposed a new method for analyzing the security of IBS schemes. FS-IBS scheme presented by Fiat and

Shamir [85] was developed from an interactive identity-based identification (IBI) scheme over zero-knowledge [94]. Fiat and Shamir [85] not only analyzed the security of FS-IBS scheme and identified the differences between identification schemes and signature schemes, but were also the first to introduce the very useful concept of the trusted center into ID-PKC. A new security proof of FS-IBS scheme was recently provided by Dodis et al. [76]. GQ-IBS scheme proposed by Guillou and Quisquater [98] is a non-pairing-based IBS scheme. Besides introducing GQ-IBS scheme, [98] also provided a solution of multiple signatures generated on GQ-IBS scheme. Similar to FS-IBS scheme, a new security proof for GQ-IBS scheme was provided by Dodis et al. [76].

After reviewing non-pairing-based IBS schemes, we next review pairing-based IBS schemes. The first IBS scheme using bilinear pairings, called SOK-IBS, was proposed by Sakai, Ohgishi and Kasahara [153] in 2000. The security of SOK-IBS scheme was not proven until 2004. Bellare et al. [20] showed that SOK-IBS scheme is secure against passive attacks, and Libert and Quisquater [118] showed that SOK-IBS scheme is secure against adaptive chosen message and identity attacks. Following the construction of identity-based encryption schemes [36] using bilinear pairings by Boneh et al., several IBS schemes using bilinear pairings were proposed. Paterson [142] proposed an IBS scheme (called Pat-IBS scheme) with simple security analysis. Compared with SOK-IBS scheme, Pat-IBS scheme is more efficient. Hess [100] and Yi [174] proposed Hess-IBS scheme and Yi-IBS scheme with respective formal security proofs. Cha and Cheon [54] described another efficient IBS scheme called CC-IBS. They defined a new formal security model for IBS schemes, called the adaptive chosen message and identity attacks. They proved that CC-IBS scheme is secure in this model. Besides the above IBS schemes, Yoon, Cheon and Kim [175] described

an IBS scheme called YCK-IBS scheme which supported batch verification [25]. The latter refers to the ability to simultaneously verify multiple signatures generated by YCK-IBS scheme. Batch verification is a very useful method in real applications such as electronic commerce. Yoon, Cheon and Kim also explained why Hess-IBS scheme and CC-IBS scheme were not suitable for batch verification in some cases. See Chapter 5 for more information. All the mentioned IBS schemes are constructed in the random oracle model, until now, building the IBS scheme in the standard model is an open problem ².

To date, the most efficient IBS scheme from the bilinear pairings is called BLMQ-IBS scheme, proposed by Barreto et al. [14]³. Some researchers have attempted to construct a framework to transfer cryptographic schemes into secure IBS schemes. Dodis et al. [76] defined a new signature, called the trapdoor standard signature (TSS), which can be transformed into a secure IBS scheme. Similar works include [20] and [111]. [20] summarized previous published IBS schemes and provided a framework for proving their security. Libert and Quisquater [119] proposed a special IBS scheme called the identity-based undeniable signature (IBUS) scheme. This is the strongest IBS scheme as it is able to provide invisibility and anonymity.

Building Non-supersingular Elliptic Curves: The aim of building non-supersingular elliptic curves suitable for pairing-based cryptosystems is to provide some explicit conditions for the Frobenius trace of elliptic curves. These conditions can help researchers to estimate whether an elliptic curve is vulnerable or se-

²It is noted that a new provably secure IBS scheme has been proposed by Paterson and Schuldt [144] after the draft of this thesis was submitted.

³The proposed IBS scheme in the random oracle model is similar to BLMQ-IBS scheme, but we offer a different security analysis. Our work is original and independent. We were aware of their work only over the Internet after submitting our IBS scheme to a conference.

cure against FR-reduction [86]. After Boneh et al. [40] suggested the use of non-supersingular elliptic curves in constructing pairing-based cryptographic schemes, researchers began to pay attention to the issue of building more non-supersingular curves suitable for pairing-based cryptosystems. This issue is one of the most active topics in ECC, and some work has been published recently [15, 16, 18, 43, 78, 89, 158]. In the initial method proposed by Miyaji et al. [130], non-supersingular elliptic curves are constructed over $\mathbb{F}_p$ whose group order is divisible by some prime r while the embedding degree k must be one of these: $\{3, 4, 6\}$. In order to construct suitable non-supersingular elliptic curves with large embedding degrees, Barreto et al. [15] extended Miyaji et al.'s method and gave some example curves when $k = 7, 11$ and 12 . They then optimized their method and presented it in [16] where additional examples were given. Dupont et al. [78] presented a method for constructing suitable non-supersingular elliptic curves with arbitrary small embedding degrees. The algorithm proposed by Brezing and Weng [43] enabled the construction of suitable elliptic curves with p significantly less than q^2 (where q is a prime), thus reducing the length of the signatures generated by the schemes proposed by [40, 41]. It is noted that [43] provides examples of non-supersingular elliptic curves where q is a prime with a low Hamming weight, making it efficient to compute pairings. Scott and Barreto [158] extended their previous method [15, 16] and gave some examples of non-supersingular elliptic curves with p approximately equal to q when $k = 3, 4$ and 6 . Barreto and Naehrig [18] gave new example curves with p approximately equal to q with the embedding degree $k = 12$. Galbraith et al. [89] developed explicit families of non-supersingular elliptic curves which included the results of [158] and provided more suitable elliptic curves. They also gave examples of suitable hyperelliptic curves [28]. Besides the methods described above, Section

IX of the book of Blake et al. [29] described a method of constructing non-supersingular elliptic curves which was originally proposed by Cocks and Pinch [29]. Their method is also called the *folklore* method and is easily constructed by using suitable elliptic curves with a low Hamming weight q when $k = 2, 4$ and 8.

The rest of this chapter is organized as follows: Section 4.2 introduces the security notions for IBS scheme and two related mathematical assumptions. Section 4.3 presents a new mathematical assumption called the generalized k -CAA and a new IBS scheme in the random oracle model. It then shows that the security of the proposed IBS scheme depends on the generalized k -CAA. Section 4.4 describes a simple method for constructing non-supersingular elliptic curves and gives two examples. Section 4.5 compares the performance of the proposed scheme and other IBS schemes at different security levels. Finally, Section 4.6 summarizes the work of this chapter.

4.2 Security Notions and Mathematical Assumptions

In this section, we introduce the security notions of IBS schemes and two mathematical assumptions which are related to the proposed schemes.

4.2.1 The Security Notion of IBS Scheme

The IBS scheme in the random oracle model requires a trusted authority (**TA**) and four algorithms: **Setup**, **Extract**, **Sign** and **Verify**. These algorithms are

used to generate the system parameters, extract the secret key associated with the user identity, sign the message using the secret key, and verify the signatures using the public key and the user identity, respectively. The security notion of the IBS scheme in the random oracle model is as follows

The Random Oracle Model: We say an IBS scheme in the random oracle model is existentially unforgeable under an adaptive chosen message and identity attack [54, 118] if no adversary $\mathcal{F}$ (here this adversary is actually an algorithm) has a non-negligible probability in polynomial time against a challenger $\mathcal{C}$ in the following game:

- **Setup:** The challenger $\mathcal{C}$ runs **Setup** to generate the public parameters and the master key. The public parameters are sent to the adversary $\mathcal{F}$ and the master key is kept secret.
- **Queries:** The adversary $\mathcal{F}$ makes the following queries:
 1. *Key extract query:* $\mathcal{F}$ adaptively issues at most q_E identities ($id_1, id_2, \dots, id_{q_E}$). $\mathcal{C}$ returns the corresponding secret keys ($S_{id_1}, S_{id_2}, \dots, S_{id_{q_E}}$) by running **Extract**.
 2. *Message hash query:* $\mathcal{C}$ computes the hash value of the messages ($m_1, m_2, \dots, m_{q_{H_2}}$) and sends them to $\mathcal{F}$.
 3. *Sign query:* Given (id_i, m_j) where $1 \leq i \leq q_E$ and $1 \leq j \leq q_S$, $\mathcal{C}$ outputs signature σ_{ij} by running **Sign** and sends them to $\mathcal{F}$.

The order of the queries is worth noting, the key extract query and the message hash query must happen before the sign query because the sign query will use the results provided by the key extract query and sign query. However, the procedure of the key extract query is

independent of the message hash query. Without loss of generality, we assume that the key extract query happens before the message hash query.

- **Output:** $\mathcal{F}$ outputs (id^*, m^*, σ^*) and wins the game if
 1. (id^*, m^*, σ^*) can pass **Verify** under the public parameters;
 2. id^* is not any of id_i and (id^*, m^*) is not any of (id_i, m_j) in the step of *Sign Query*.

Otherwise, $\mathcal{F}$ stops and outputs failure.

Let ϵ denote the probability that the adversary $\mathcal{F}$ wins the above game. We thus define:

Definition 4.1. *In the random oracle model, an algorithm $\mathcal{F}$ can $(t, q_{H_2}, q_E, q_S, \epsilon)$ -break an IBS scheme if $\mathcal{F}$ outputs a forgery with probability at least ϵ by running in time at most t , making at most q_{H_2} queries to the hash oracle, q_E extract queries, q_S signature queries. An IBS signature scheme is $(t, q_{H_2}, q_E, q_S, \epsilon)$ -existential unforgeable under an adaptive chosen message and identity attack if no known efficient polynomial time algorithm $(t, q_{H_2}, q_E, q_S, \epsilon)$ -breaks it.*

4.2.2 Related Mathematical Assumption

The Collusion of Attack Algorithm with k Traitors: The security of ZSS short signature scheme [182] depends on this mathematical assumption: there is no known efficient polynomial time algorithm for the *Collusion of Attack Algorithm*

with k Traitors (k -CAA) [129]. G_1 and G_2 are two additive groups described in Section 2.6.1 of Chapter 2. The definition of k -CAA is as follows [129]:

Definition 4.2. For a known $k \in \mathbb{Z}$ and an unknown $x \in \mathbb{Z}_p^*$, k -CAA is an algorithm which can compute $Q = \frac{1}{x+g}P_1$ from given $(g_1, g_2, \dots, g_k \in \mathbb{Z}_p^*, P_1 \in G_1, xP_1, \frac{1}{x+g_1}P_1, \frac{1}{x+g_2}P_1, \dots, \frac{1}{x+g_k}P_1)$, where $g \in \mathbb{Z}_p^*$ and not any of $\{g_1, g_2, \dots, g_k\}$.

Definition 4.3. If the tuple $(g_1, g_2, \dots, g_k, P_1, xP_1, \frac{1}{x+g_1}P_1, \frac{1}{x+g_2}P_1, \dots, \frac{1}{x+g_k}P_1)$ is given, an algorithm $\mathcal{A}$ can output $Q = \frac{1}{x+g}P_1$ for some $g \notin \{g_1, g_2, \dots, g_k\}$ in at most time t with the possibility of at least ϵ . The advantage of the algorithm $\mathcal{A}$ solving k -CAA is

$$\text{Adv}(\mathbf{k} - \mathbf{CAA})_{\mathcal{A}} = \Pr \left[\begin{array}{l} x \in \mathbb{Z}_p^* \text{ and } P_1 \in G_1; \\ g_1, g_2, \dots, g_k \in \mathbb{Z}_p^*, g \notin \{g_1, g_2, \dots, g_k\}; \\ \mathcal{A}(P_1, xP_1, \frac{1}{x+g_1}P_1, \frac{1}{x+g_2}P_1, \dots, \frac{1}{x+g_k}P_1) = \frac{1}{x+g}P_1 \end{array} \right]$$

We say the algorithm $\mathcal{A}$ (t, ϵ) -breaks k -CAA if $\mathcal{A}$ runs in time at most t and $\text{Adv}(\mathbf{k} - \mathbf{CAA})_{\mathcal{A}} \geq \epsilon$. Otherwise, we say that k -CAA is (t, ϵ) -hard if $\mathcal{A}$ runs in time at most t and $\text{Adv}(\mathbf{k} - \mathbf{CAA})_{\mathcal{A}} < \epsilon$.

4.3 An Efficient IBS Scheme in The Random Oracle Model

In this section, we first introduce a new complexity assumption called Generalized k -CAA. We then present a new IBS scheme in the random oracle model, whose security depends on this new complexity assumption.

4.3.1 Generalized k -CAA

Before introducing the proposed IBS scheme in the random oracle model, we first propose a new complexity assumption. Because k -CAA is viewed as a special case of the new complexity assumption, we hereby call it Generalized k -CAA. Its definition is as follows:

Definition 4.4. For a known $k \in \mathbb{Z}$ and an unknown $x \in \mathbb{Z}_p^*$, k is the product of two integers m and n , Generalized k -CAA is an algorithm which computes $\frac{f}{x+g}P_1$ from a given tuple $(f_1, f_2, \dots, f_m, g_1, g_2, \dots, g_n \in \mathbb{Z}_p^*, P_1 \in G_1, xP_1, \frac{f_1}{x+g_1}P_1, \frac{f_2}{x+g_1}P_1, \dots, \frac{f_m}{x+g_1}P_1, \frac{f_1}{x+g_2}P_1, \dots, \frac{f_m}{x+g_{n-1}}P_1, \frac{f_1}{x+g_n}P_1, \dots, \frac{f_m}{x+g_n}P_1)$, where $f, g \in \mathbb{Z}_p^*, f \notin \{f_1, f_2, \dots, f_m\}$ and $g \notin \{g_1, g_2, \dots, g_n\}$.

Definition 4.5. If the tuple $(f_1, f_2, \dots, f_m, g_1, g_2, \dots, g_n \in \mathbb{Z}_p^*, P_1 \in G_1, xP_1, \frac{f_1}{x+g_1}P_1, \frac{f_2}{x+g_1}P_1, \dots, \frac{f_m}{x+g_1}P_1, \frac{f_1}{x+g_2}P_1, \dots, \frac{f_m}{x+g_{n-1}}P_1, \frac{f_1}{x+g_n}P_1, \dots, \frac{f_m}{x+g_n}P_1)$ is given, an algorithm $\mathcal{A}$ outputs $Q = \frac{f}{x+g}P_1$ for $f \notin \{f_1, f_2, \dots, f_m\}$ and $g \notin \{g_1, g_2, \dots, g_n\}$ in at most time t with a possibility of at least ϵ . The advantage the algorithm $\mathcal{A}$ solving generalized k -CAA is

$$Adv(\mathbf{Genk-CAA})_{\mathcal{A}} = Pr \left[\begin{array}{l} x \in \mathbb{Z}_p^*, P_1 \in G_1 \text{ and } k = m \times n; \\ f_1, f_2, \dots, f_m \in \mathbb{Z}_p^* \text{ and } f \notin \{f_1, f_2, \dots, f_m\}; \\ g_1, g_2, \dots, g_n \in \mathbb{Z}_p^* \text{ and } g \notin \{g_1, g_2, \dots, g_k\}; \\ \mathcal{A}(P_1, xP_1, \frac{f_1}{x+g_1}P_1, \frac{f_1}{x+g_2}P_1, \dots, \frac{f_m}{x+g_n}P_1) = \frac{f}{x+g}P_1 \end{array} \right]$$

We say the algorithm $\mathcal{A}$ (t, ϵ) -breaks Generalized k -CAA if $\mathcal{A}$ runs in time at most t and $Adv(\mathbf{Gen.k-CAA})_{\mathcal{A}} \geq \epsilon$. Otherwise, we say that k -CAA is (t, ϵ) -hard if $\mathcal{A}$ runs in time at most t and $Adv(\mathbf{Gen.k-CAA})_{\mathcal{A}} < \epsilon$.

Lemma 4.1. Generalized k -CAA is at least polynomially equivalent to k -CAA.

Proof. Let $f_i = 1$, Generalized k -CAA is transformed into n -CAA. Thus, k -CAA can be seen as a special case of Generalized k -CAA. We make an assumption: There is a known polynomial time algorithm using some oracle which has solved Generalized k -CAA. For a given instance of n -CAA as follows: $(g, g_1, g_2, \dots, g_n \in \mathbb{Z}_p^*, P_1 \in G_1, xP_1, \frac{1}{x+g_1}P_1, \frac{1}{x+g_2}P_1, \dots, \frac{1}{x+g_n}P_1)$ where $g \notin \{g_1, g_2, \dots, g_n\}$. We can convert it to an instance of Generalized k -CAA by taking m f_i where $k = m \times n$, and compute $\frac{f}{x+g}P_1$ by the algorithm using the oracle for $f \notin \{f_1, f_2, \dots, f_m\}$. Given f and $\frac{f}{x+g}P_1$, it is easy to solve n -CAA by computing $\frac{1}{x+g}P_1$. Based on the above description, we can make a conclusion that the difficulty of Generalized k -CAA is at least polynomially equivalent to k -CAA. From the assumption of k -CAA introduced in Section 4.2.2, there is no known efficient polynomial time algorithm for solving n -CAA until now. Therefore, no known polynomial time polynomial time algorithm can solve Generalized k -CAA. $\square$

4.3.2 The Proposed IBS Scheme

In the existing IBS schemes from the bilinear pairings [142, 54, 100, 174, 175], extracting the secret key from the master key and the user identity requires a special hash function called the *MapToPoint* function [40] which maps user identity, messages or random information, into a point of elliptic curves. Recently, Mitsunari et al. [129] proposed another method for generating the secret key S_{id} from the master key and the user identity. Their method is described as follows: Let x be the master key and id be the user identity where $x, id \in \mathbb{Z}_p^*$. For $P_1 \in G_1$, the corresponding secret key can be produced by $S_{id} = \frac{1}{x+id}P_1$ (or $S_{id} = \frac{1}{x+H_1(id)}P_1$ where H_1 is a regular collision resistant hash function). Using this method of generating the secret key, a new IBS scheme without the

MapToPoint function is constructed.

Let E be a supersingular elliptic curve of order q . $\mathbb{F}_p$ is a finite field. $G_1 = E(\mathbb{F}_p)$, $P_1, P_2 \in G_1$. $\hat{e}_t(P_1, P_2)$ represents the modified Tate pairing. The proposed IBS scheme is described as follows:

- **Setup:** Given a security parameter $\kappa \in \mathbb{N}$, **TA** first chooses suitable system parameters such as G_1, G_3 , then randomly picks $P_1 \in G_1$ and $x \in \mathbb{Z}_p^*$. Next, it computes the public key $P_{pub} = xP_1$ and precomputes $\omega = \hat{e}_t(P_1, P_1)$. x is the master key. The public parameter is $(p, G_1, G_3, \hat{e}_t, P_1, P_{pub}, \omega, H_1, H_2)$, where the hash function H_1 is defined as $\{0, 1\}^* \rightarrow \mathbb{Z}_p^*$ and the hash function H_2 is defined as $\{0, 1\}^* \times G_3 \rightarrow \mathbb{Z}_p^*$.
- **Extract:** For a given identity $id \in \mathbb{Z}_p^*$, **TA** computes the secret key $S_{id} = \frac{1}{x + H_1(id)} P_1$. Note if $x + H_1(id) \equiv 0 \pmod{p}$, then abort x and return **Setup** such that a new master key x is generated.
- **Sign:** Given the secret key S_{id} and the message $m \in \{0, 1\}^*$, the signer chooses a random element s from $\mathbb{Z}_p^*$ and computes $r = \omega^s$, $u = H_2(m, r)$, $V = (u + s)S_{id}$. The signature pair (r, V) is sent to the verifier.
- **Verify:** Given the public parameters $(p, G_1, G_3, \hat{e}_t, P_1, P_{pub}, \omega, H_1, H_2)$, a user identity id and a signature pair (r, V) on the message m , the verifier computes $u = H_2(m, r)$, and accepts the signature if

$$\omega^u r = \hat{e}_t(P_{pub} + H_1(id) \cdot P_1, V).$$

Note that **Extract** is only executed once for every identity. The procedure

of the verification is deduced as follows:

$$\hat{e}_t(P_{pub} + H_1(id) \cdot P_1, V) = \hat{e}_t((x + H_1(id))P_1, S_{id})^{(u+s)} = \omega^{u+s} = \omega^u r$$

4.3.3 Security Analysis of The Proposed IBS Scheme

The forking lemma [146, 149] is necessary for proving the security of existing IBS schemes [54, 100, 174, 175]. However, it does not yield tight security reductions [118]. Recently, some signature schemes [32, 182, 177] have been proven to be secure under the adaptive chosen message attack, without the forking lemma being used in the proofs. In this section, we adopt the methods of these authors to prove the security of the proposed IBS scheme under adaptive chosen message and identity attacks in the random oracle model.

To prove that the security of the proposed scheme depends on Generalized k -CAA, the following theorem is given:

Theorem 4.1. *In the random oracle model, if an algorithm $\mathcal{F}(t, q_{H_2}, q_E, q_S, \epsilon)$ -breaks the proposed IBS scheme under adaptive chosen message and identity attacks, then another (t', ϵ') -algorithm $\mathcal{C}$ exists that is able to solve Generalized k -CAA, where $t' = t$, $q_S \leq q_{H_2}$, $k = q_E \cdot q_S$ and $\epsilon' = \frac{(p-q_S) \cdot q_E^{q_E} \cdot q_S^{q_E \cdot q_S}}{p^{q_E+1} \cdot q_{H_1}^{q_E} \cdot q_{H_2}^{q_E \cdot q_S}} \cdot \epsilon$.*

Proof. Suppose that an algorithm $\mathcal{F}(t, q_{H_2}, q_E, q_S, \epsilon)$ -breaks the proposed IBS scheme under adaptive chosen message and identity attacks. We expect to construct an algorithm $\mathcal{C}$ to solve Generalized k -CAA from $\mathcal{F}$. Given a tuple $(f_1, f_2, \dots, f_m, g_1, g_2, \dots, g_n \in \mathbb{Z}_p^*, P_1 \in G_1, xP_1, \frac{f_1}{x+g_1}P_1, \frac{f_2}{x+g_1}P_1, \dots, \frac{f_m}{x+g_1}P_1, \frac{f_1}{x+g_2}P_1, \dots, \frac{f_m}{x+g_{n-1}}P_1, \frac{f_1}{x+g_n}P_1, \dots, \frac{f_m}{x+g_n}P_1)$, $\mathcal{C}$ has an ability to output $\frac{f}{x+g}P_1$

for $f \notin \{f_1, f_2, \dots, f_m\}$ and $g \notin \{g_1, g_2, \dots, g_n\}$. In the following simulation, $\mathcal{F}$ and $\mathcal{C}$ play the role of the adversary and the challenger, respectively. $\mathcal{F}$ will interact with $\mathcal{C}$ as follows:

- **Setup:** $\mathcal{C}$ runs **Setup** to obtain the public parameters $(p, G_1, G_3, \hat{e}_t, P_1, Q, \omega, H_2)$ where $Q = xP_1$. $x \in \mathbb{Z}_p^*$ is the master key. The public parameters are sent to $\mathcal{F}$.
- **Queries:** $\mathcal{F}$ issues the following queries for the identities $(id_1, id_2, \dots, id_{q_E})$ and the messages $(m_1, m_2, \dots, m_{q_S})$ (without loss of generality, we assume that, for any key extract query or signature query involving an identity, an identity query was previously issued for the same identity):
 1. *Preparing query:* $\mathcal{C}$ prepares a response set $\{d_1, d_2, \dots, d_{q_{H_1}}\}$ of the identity hash oracle queries on id_i where $d_i \in \mathbb{Z}_p^*$. The response set includes a subset $\{e_1, e_2, \dots, e_{q_E}\}$. Let $e_{i'}$ denote the element of the subset and $e_{i'}$ is distributed randomly in the response set. It is noted that q_{H_1} should not be less than q_E).
 2. *Identity hash query:* $\mathcal{F}$ makes a hash oracle query on id_i for $1 \leq i \leq q_{H_1}$, $\mathcal{C}$ sends d_i to $\mathcal{F}$ as the response of the hash query on id_i .
 3. *Key extract query:* $\mathcal{F}$ makes an extract query on d_i (where d_i represents the identity id_i) for $1 \leq i \leq q_E$. if $d_i = e_{i'}$, $\mathcal{C}$ responds $S_{id_i} = \frac{1}{x+e_{i'}}P_1$ as the response (Note that x is a unknown value, computing $S_{id_i} = \frac{1}{x+e_{i'}}P_1$ is based on some probability of computing the value of S_{id_i} which is analyzed in the discussion below). Otherwise, $\mathcal{C}$ stops the simulation and report failure.
 4. *Message hash query:* For any given message m_j ($1 \leq j \leq q_{H_2}$), $\mathcal{C}$

constructs a L_1 -list of tuple $\langle m_j, r_j, u_j, s_j \rangle$ for responding to the hash queries of $\mathcal{F}$. When $\mathcal{F}$ sends a hash query for the message m_j , $\mathcal{C}$ picks one random element s_j from $\mathbb{Z}_p^*$, and then computes $r_j = \omega^{s_j}$ and $u_j = H_2(m_j, r_j)$. It is noted that $s_i + u_i \neq s_j + u_j$ when $i \neq j$. u_j is sent to $\mathcal{F}$ as the response of the hash query on the message m_j . Simultaneously, $\mathcal{C}$ constructs another L_2 -list $\{h_1, h_2, \dots, h_{q_{H_2}}\}$ where $h_j = u_j + s_j$.

5. *Sign query*: For any given identity-message pair (id_i, m_j) where $1 \leq i \leq q_E$ and $1 \leq j \leq q_{H_2}$, $\mathcal{C}$ first runs the hash query algorithm to check whether m_j appears in the L_1 -list. If not, $\mathcal{C}$ stops the simulation and reports failure. Otherwise, $\mathcal{C}$ obtains the corresponding r_j, u_j and s_j from L_1 and computes

$$V_{ij} = (u_j + s_j)S_{id_i} = \frac{u_j + s_j}{x + e_{i'}} P_1.$$

$\mathcal{C}$ searches h_k from L_2 -list such that $h_k = u_j + s_j$ (where $1 \leq j \leq q_S$, $1 \leq k \leq q_{H_2}$, $q_S \leq q_{H_2}$), then the pair $(r_j, \frac{h_k}{x + e_{i'}} P_1)$ is viewed as the signature on the message m_j for the user id_i from $\mathcal{F}$'s point of view. $\mathcal{C}$ returns it to $\mathcal{F}$ as the response of the signing oracle.

- **Output**: Finally, $\mathcal{F}$ outputs a pair (r^*, V^*) on the message m^* for the user id^* , and accepts it if the following conditions are satisfied:

1. id^* is not any of id_i and (id^*, m^*) is not any of (id_i, m_j) in the *Sign Query* phase;
2. (id^*, m^*, r^*, V^*) can successfully pass **Verify** under the public parameters.

Suppose $r^* = \omega^{s^*}$ and $H_2(m^*, r^*) = u^* \in \mathbb{Z}_p^*$ such that $h^* = u^* + s^* \notin \{h_1, h_2, \dots, h_{q_{H_2}}\}$, where s^* and u^* are two random elements in $\mathbb{Z}_p^*$. Since the output (id^*, m^*, r^*, V^*) of $\mathcal{F}$ can pass **Verify**'s check, we obtain this:

$$\hat{e}_t(Q + H_1(id^*) \cdot P_1, V^*) = \omega^{u^*} r^* \Rightarrow \hat{e}_t(P_1, V^*)^{(x+H_1(id^*))} = \hat{e}_t(P_1, P_1)^{(u^*+s^*)}$$

Therefore, $V^* = \frac{u^*+s^*}{x+H_1(id^*)} P_1 = \frac{h^*}{x+H_1(id^*)} P_1$. From $\mathcal{C}$'s point of view, $V^* = \frac{h^*}{x+H_1(id^*)} P_1$ is viewed as the solution of Generalized k -CAA. The reasoning is as follows: when $m = q_S$ and $n = q_E$, namely $k = q_E \cdot q_S$, $\mathcal{C}$ can compute V^* from the known tuple $(h_1, h_2, \dots, h_{q_S}, e_1, e_2, \dots, e_{q_E} \in \mathbb{Z}_p^*, P_1 \in G_1, xP_1, \frac{h_1}{x+e_1}P_1, \frac{h_2}{x+e_1}P_1, \dots, \frac{h_{q_S}}{x+e_1}P_1, \frac{h_1}{x+e_2}P_1, \dots, \frac{h_{q_S}}{x+e_{q_E-1}}P_1, \frac{h_1}{x+e_{q_E}}P_1, \dots, \frac{h_{q_S}}{x+e_{q_E}}P)$ where h_i is from the response of the message hash query on the message m_i and the pair (m_i, e_i) is random by $\mathcal{F}$'s adaptive choices.

Since the message hash function behaves as a random oracle, $\mathcal{F}$ is not sure whether $\mathcal{C}$ is a simulator or a real attacker. The running time t' of $\mathcal{C}$ is the same as t of $\mathcal{F}$. In the *Identity query* phase, $\mathcal{C}$ stops the simulation and reports failure when $d_i \neq e_i$. The probability of this event is that $\mathcal{C}$ successfully computes S_{id_i} is $\frac{q_E}{q_{H_1}}$. The probability that $\mathcal{C}$ successfully computes S_{id_i} is $\frac{q_E}{p \cdot q_{H_1}}$. For all identity oracle queries, $\mathcal{C}$ will not fail with the probability $(\frac{q_E}{p \cdot q_{H_1}})^{q_E}$. In the *Sign Query* phase, $\mathcal{C}$ stops the simulation and reports failure only when m_j is not in the L_1 . The probability that this event does not happen is $\frac{q_S}{q_{H_2}}$. For all the q_S sign queries, $\mathcal{C}$'s probability of success is $(\frac{q_S}{q_{H_2}})^{q_E \cdot q_S}$. Furthermore, the probability of another independent event, $h^* = u^* + s^* \notin \{h_1, h_2, \dots, h_{q_{H_2}}\}$, is $(1 - \frac{q_S}{p})$. Hence, $\mathcal{C}$'s successful probability ϵ' is $\frac{(p-q_S) \cdot q_E^{q_E} \cdot q_S^{q_S}}{p^{q_E+1} \cdot q_{H_1}^{q_E} \cdot q_{H_2}^{q_S}} \cdot \epsilon$.

□

Remark 4.1. the proposed IBS scheme can be modified such that shorter signa-

tures are provided. In the modified version, (h, V) is sent to the verifier as the signature, instead of (r, V) . We note that the security of the modified scheme is the same as the original scheme. Nevertheless, the modified scheme is not suitable for carrying out batch verification, which will be introduced in the next chapter.

4.4 A Method for Building Non-Supersingular Elliptic Curves

Suppose that an elliptic curve E over a finite field $\mathbb{F}_p$ has an order q , which is the product of a small integer h and a large prime r . Let k denote the embedding degree, *i.e.* r divides $(p^k - 1)$ but does not divide $(p^l - 1)$ when $1 < l < k$. Let D and t denote the discriminant and the trace of a Frobenius endomorphism on E , respectively. The following equations should be satisfied:

$$p + 1 - t \equiv 0 \pmod{r} \quad (4.1)$$

$$p^k - 1 \equiv 0 \pmod{r} \quad (4.2)$$

$$t^2 - 4p = Dv^2 \text{ (where } v \in \mathbb{Z}) \quad (4.3)$$

From the above three equations, Cocks and Pinch [29] made an important discovery. Substituting $p \equiv t - 1 \pmod{r}$, which is easily derived from Equation

4.1, for p in Equation 4.2, the following equation is obtained:

$$(t - 1)^k \equiv 1 \pmod{r} \quad (4.4)$$

Thus, $(t - 1)$ is viewed as a primitive k -th root of unity modulo r . This discovery is very useful for building non-supersingular elliptic curves that are suitable for our purposes. A prior example was by Brezing and Weng [43], who used it to construct non-supersingular elliptic curves suitable for pairing-based cryptography.

In [44], we observed that $(1 - t)^k \equiv 1 \pmod{r}$ is correct when k is even, and concluded that $(1 - t)$ is a primitive k -th root of unity modulo r when k is even. However, this conclusion is not precise. Galbraith [87] pointed out that $(1 - t)$ is not always a primitive k -th root of unity modulo r when k is even. Suppose $k = 2i$ and i is odd. Let $a = (t - 1)$ be a primitive k -th root of unity modulo r . Then $a^k \equiv 1 \pmod{r}$ but $a^i \equiv -1 \pmod{r}$, so $(-a)^i \equiv 1 \pmod{r}$. Thus, $(1 - t)$ (or $-a$) should be a primitive $\frac{k}{2}$ -th root of unity modulo r when $k = 2i$ and i is odd. When $k = 2i$ and i is even, we obtain $(-a)^k \equiv 1 \pmod{r}$ and $(-a)^i = a^i \equiv -1 \pmod{r}$. Thus $(1 - t)$ (or $-a$) should be a primitive k -th root of unity modulo r only when $k = 2i$ and i is even. The following lemma summarizes the description above.

Lemma 4.2. *An elliptic curve E over $\mathbb{F}_p$ with embedding degree k has an order q , which is the product of a small integer h and a large prime r . Let t denote the trace of Frobenius. When $k = 2i$ and i is even, both $(t - 1)$ and $(1 - t)$ are the primitive k -th roots of unity modulo r .*

We modify Algorithm IX.4 of [29] such that more suitable non-supersingular

Algorithm 1: Constructing ordinary elliptic curves	
Input:	a positive integer $k = 2i$ where i is even, a prime r with low Hamming weight and a negative squarefree integer D
Output:	p, t such that an elliptic curve is generated by the discriminant D over $\mathbb{F}_p$ with $(p + 1 - t)$ points where r divides $(p + 1 - t)$ and $(p^k - 1)$.

1. Choose a primitive k -th root of unity f in $\mathbb{F}_r$;
2. Choose an integer $u = \frac{1-g}{2}$;
3. If $(\gcd(u, D) = 1)$, then goto Step 5;
4. Else goto Step 1 and choose another g ;
5. Choose an integer $v_0 \equiv \pm \frac{u-1}{\sqrt{D}} \pmod{r}$;
6. $j \leftarrow 0$;
7. Do
8. $p \leftarrow u^2 - D(v_0 + jr)^2$;
9. $j \leftarrow 1$;
10. Until p is a prime;
11. $t = 2u$;
12. Return p and t ;

elliptic curves are constructed based on Lemma 4.5. The modified version is shown in Algorithm 1. Compared with Algorithm IX.4 of [29], we replace $u = \frac{1+g}{2} \pmod{r}$ with $u = \frac{1-g}{2} \pmod{r}$.

Example 1 and Example 2 give two suitable non-supersingular elliptic curves when the embedding degree is 4 and 8, respectively. Both of them satisfy the form of Equation 2.2. They are generated by the Complexity Multiplication method (see Chapter VIII of [28]). h represents the cofactor. a_4 is set as -3 because computing the pairings on such curves is more efficient (see Appendix A of [113]). Algorithm 1 provides an alternative method for constructing suitable non-supersingular elliptic curves when $k = 2i$ and i is even.

Example 1:

$$k = 4;$$

$$D = 34621;$$

$$r = 3138550867693340381917894711603833208051177722232017256453;$$

(192 bits)

$$p = 77692240534303589186905820321978401394804969000475646913129$$

$$64755805871852605772263889006579669575724051555197829606350$$

$$112740811222226836733067592930535269; \quad (512 \text{ bits})$$

$$q = 77692240534303589186905820321978401394804969000475646913129$$

$$64755805871852605772263889006579669581315233969050243536009$$

$$026361583792700997809605287589896446;$$

$$h = 24754175990591176023634143856187378260045036678988045297406$$

$$98404849474883215731894707121688974182$$

$$t = -5591182413852413929658913620772570474161076537694659361176;$$

$$a_4 = -3;$$

$$a_6 = 387256916318684153327442182028602361692894077811345794682$$

$$101994660813688119386294835087955170471857222783882966646$$

$$9377009934062549421187519849038821993888;$$

Example 2:

$$\begin{aligned}
k &= 8; \\
D &= 50213; \\
r &= 1347997333357531989733350754350981533681857221127028624055 \\
&\quad 1805132801; \quad (224 \text{ bits}); \\
p &= 1185314388447057279493758999319166551810712567631582892941 \\
&\quad 8083209982992840038089680442454228181685443822929462557269 \\
&\quad 628947827273170813123400658319248233269 \quad (512 \text{ bits}); \\
q &= 1185314388447057279493758999319166551810712567631582892941 \\
&\quad 8083209982992840038089680442457029931824075551562039490035 \\
&\quad 291828228668615210110814131894436971412; \\
h &= 8793150840252248539869120430365836269975638009687395336329 \\
&\quad 33466957672502023928355537812; \\
t &= -280175013863172863257693276566288040139544439698741347357 \\
&\quad 5188738142; \\
a_4 &= -3; \\
a_6 &= 88767383962872092275837820165601528215876891111951894629 \\
&\quad 36177745759267141249641242002152506016744623003426796799 \\
&\quad 253606180917096745937267167282683072524232
\end{aligned}$$

4.5 Performance

Verifying signatures is the most time consuming procedure in the IBS schemes.

In this section, we implement the proposed IBS schemes and other IBS schemes

using bilinear pairings [153, 142, 54] on supersingular and non-supersingular elliptic curves over prime fields at different security levels. Following that, their performances are compared, especially the performance of verifying signatures. The experimental results show that the proposed IBS scheme is more efficient than other IBS schemes [153, 142, 54, 100, 174, 175] in the random oracle model at the 1024 bits and 2048 bits security levels.

When the characteristic of the finite field is either 2 or 3, only a small number of suitable supersingular elliptic curves exist. We choose those supersingular (or non-supersingular) elliptic curves over prime fields. The equations of these elliptic curves follow Equation 2.1 in Chapter 2. In order to minimize the effects of exponentiating a pairing, the value of the embedding degree should be as small as possible. Table 4.1 shows the timings for the basic cryptographic primitives at different security levels. For example, $E_{ss2}^{1024}(\mathbb{F}_{512})$ refers to a supersingular elliptic curve over $\mathbb{F}_{512}$ with embedding degree 2 which is secure at the 1024 bits level. Similarly, $E_{ns4}^{2048}(\mathbb{F}_{512})$ refers to a non-supersingular elliptic curve over $\mathbb{F}_{512}$ with embedding degree 4 which is secure at 2048 bits level. The first row shows the timing of the primitives on a supersingular elliptic curve $E_{ss2}^{1024}(\mathbb{F}_{512})$. To resist index-calculus attack [138] and Pohlig-Hellman attack [138] at the 1024 bits security level, q is a 160 bits prime and p is a 512 bits prime. In such a curve, $G_1 = E(\mathbb{F}_{512})$, $G_2 = E(\mathbb{F}_{512^2})$. The timings for cryptographic primitives over $\mathbb{F}_p$, G_1 and G_3 are shown in the first row of Table 4.1. I , M_{G_1} , M_{G_2} , $H_{M_{G_1}}$, $H_{M_{G_2}}$, Pa , A_{G_1} , A_{G_2} , M_{G_3} , D_{G_3} and E_{G_3} denote the cost of computing an inverse operation over $\mathbb{F}_p$, a scalar multiplication in G_1 , a scalar multiplication in G_2 , a *MapToPoint* function [36] on G_1 , a *MapToPoint* function [36] on G_2 , a pairing, a point addition on G_1 , a point addition on G_2 , a multiplication on G_3 , a division on G_3 and an exponentiation on G_3 , respectively. The implementation

Table 4.1: Timings of the cryptographic primitives

	I (ms)	M_{G_1} (ms)	M_{G_2} (ms)	$H_{M_{G_1}}$ (ms)	$H_{M_{G_2}}$ (ms)	Pa (ms)	A_{G_1} (ms)	A_{G_2} (ms)	M_{G_3} (ms)	D_{G_3} (ms)	E_{G_3} (ms)
$E_{ss2}^{1024}(\mathbb{F}_{512})$	0.02	5.01	10.00	15.45	N/A	20.28	0.03	N/A	0.01	0.05	1.92
$E_{ns2}^{1024}(\mathbb{F}_{512})$	0.02	5.01	10.00	15.08	2.21	21.72	0.05	0.17	0.02	0.06	2.16
$E_{ns6}^{1024}(\mathbb{F}_{157})$	0.02	1.44	10.17	2.21	108.87	136.19	0.01	1.31	0.30	0.79	41.84
$E_{ns4}^{2048}(\mathbb{F}_{512})$	0.02	5.53	10.00	13.28	6.06	89.56	0.03	0.15	0.06	0.20	11.00
$E_{ns8}^{4096}(\mathbb{F}_{512})$	0.02	6.61	10.00	13.35	20.07	361.07	0.05	0.50	0.27	0.70	56.61

of these primitives was performed using Miracl [162] on a Pentium IV 2.26GHz with 256M RAM. The parameters of $E_{ns4}^{2048}(\mathbb{F}_{512})$ and $E_{ns8}^{4096}(\mathbb{F}_{512})$ are shown in Example 1 and 2. Other related parameters are given in Appendix C.

The results in Table 4.1 show that the cost of I , A_{G_1} , A_{G_2} are trivial in comparison with other primitives. Thus, they are usually omitted in the following analysis unless specifically mentioned.

Hess [100] has shown that, in terms of efficiency, Hess-IBS was at an advantage compared to other schemes (SOK-IBS scheme [153], Pat-IBS scheme [142], CC-IBS scheme [54]). Hence, only Hess-IBS is considered in the Table 4.2 - 4.7. Besides [153, 142, 54, 100], Yi-IBS [174] provides the shortest signature. The IBS scheme introduced by Yoon et al. (YCK-IBS) [175] is included in Table 4.2 - 4.7. Thus, Table 4.2 lists the main primitives required by the proposed IBS scheme, Hess-IBS scheme, Yi-IBS scheme and YCK-IBS scheme. Table 4.3 - 4.7 shows the detailed timings for all the IBS schemes on different security level.

- **Case 1:** When the elliptic curve is $E_{ss2}^{1024}(\mathbb{F}_{512})$, we can observe from Table 4.2 and 4.3 that the proposed IBS scheme has the shortest running time for extracting a secret key. In the **Sign** phase, both the proposed IBS scheme and Hess-IBS require $(1M_{G_1} + 1E_{G_3})$ which is faster than $3M_{G_1}$ in Yi-IBS

Table 4.2: The comparison of the proposed schemes and other IBS schemes

Schemes	Precomputation	Setup	Extract	Sign	Verify
The proposed IBS Scheme	$1Pa$	$1M_{G_1}$	$1I + 1M_{G_1}$	$1M_{G_1} + 1E_{G_3}$	$1M_{G_1} + 1E_{G_3} + 1Pa$
Hess-IBS	$1Pa$	$1M_{G_1}$	$1H_{M_{G_1}} + 1M_{G_1}$	$1M_{G_1} + 1E_{G_3}$	$1H_{M_{G_1}} + 2Pa + 1E_{G_3}$
Yi-IBS	N/A	$1M_{G_1}$	$1H_{M_{G_1}} + 1M_{G_1}$	$3M_{G_1}$	$1H_{M_{G_1}} + 1M_{G_1} + 2Pa$
YCK-IBS	N/A	$1M_{G_1}$	$1H_{M_{G_1}} + 1M_{G_1}$	$1H_{M_{G_1}} + 3M_{G_1}$	$1H_{M_{G_1}} + 1M_{G_1} + 2Pa$

Table 4.3: Timings of the proposed schemes and other IBS schemes on $E_{ss2}^{1024}(\mathbb{F}_{512})$

Schemes	Precomputation (ms)	Setup (ms)	Extract (ms)	Sign (ms)	Verify (ms)
the proposed IBS scheme	20.28	5.01	5.03	6.93	27.21
Hess-IBS	20.28	5.01	20.37	6.93	57.84
Yi-IBS	N/A	5.01	20.37	15.03	60.93
YCK-IBS	N/A	5.01	20.37	30.39	60.93

and $(1H_M + 3M_{G_1})$ in YCK-IBS. In the **Verify** phase, the proposed IBS scheme $(1M_{G_1} + 1Pa + 1E_{G_3})$ is more efficient than Hess-IBS $(1H_M + 1E_{G_3} + 2Pa)$, Yi-IBS $(1H_M + 1M_{G_1} + 2Pa)$ and YCK-IBS $(1H_M + 1M_{G_1} + 2Pa)$. From the results of Table 4.3, the verification of the proposed IBS scheme has an improvement of approximately 53% over Hess-IBS scheme, and 55% over Yi-IBS scheme and YCK-IBS scheme. We note that Hess-IBS scheme can reduce the **Verify** phase by one pairing computation when the same identities occur frequently [100]; however, computing two pairings is still necessary in the first verification. Therefore, the proposed IBS scheme provides the fastest verification among all of the IBS schemes in the random oracle model [153, 142, 54, 100, 174, 175].

- **Case 2:** When the elliptic curve is $E_{ns2}^{1024}(\mathbb{F}_{512})$, the timing for this case

Table 4.4: Timings of the proposed schemes and other IBS schemes on $E_{ns2}^{1024}(\mathbb{F}_{512})$

Schemes	Precomputation (ms)	Setup (ms)	Extract (ms)	Sign (ms)	Verify (ms)
the proposed IBS scheme	21.72	5.01	5.03	7.17	23.9
Hess-IBS	21.72	5.01	20.09	7.17	60.68
Yi-IBS	N/A	5.01	20.09	15.03	63.53
YCK-IBS	N/A	5.01	20.09	30.11	63.53

Table 4.5: Timings of the proposed schemes and other IBS schemes on $E_{ns6}^{1024}(\mathbb{F}_{157})$

Schemes	Precomputation (ms)	Setup (ms)	Extract (ms)	Sign (ms)	Verify (ms)
the proposed IBS scheme	136.19	1.44	1.46	43.28	178.05
Hess-IBS	136.19	1.44	3.65	43.28	316.43
Yi-IBS	N/A	1.44	3.65	4.32	276.03
YCK-IBS	N/A	1.44	3.65	6.53	276.03

is similar to Case 1. The reason is that the embedding degree is 2. Thus, the exponentiation computation on G_3 is not time consuming. the proposed IBS scheme still provides the most efficient **Setup**, **Extract**, **Sign** and **Verify**. In addition, the results show that computing pairings on supersingular elliptic curves is a little faster than that on non-supersingular elliptic curves.

- **Case 3:** When the elliptic curve is $E_{ns6}^{1024}(\mathbb{F}_{157})$, the embedding degree is 6 and the order of the curve is reduced as a 157 bits prime, so that 1024 bits security is maintained. From the results in Table 4.6, the timing for computing a scalar multiplication on G_1 is reduced while the timings

Table 4.6: Timings of the proposed schemes and other IBS schemes on $E_{ns4}^{2048}(\mathbb{F}_{512})$

Schemes	Precomputation (ms)	Setup (ms)	Extract (ms)	Sign (ms)	Verify (ms)
the proposed IBS scheme	89.56	5.53	5.55	16.53	100.58
Hess-IBS	89.56	5.53	18.81	16.53	203.40
Yi-IBS	N/A	5.53	18.81	16.59	197.93
YCK-IBS	N/A	5.53	18.81	29.87	197.93

for computing an exponentiation on G_3 and computing a pairing are increased. Yi-IBS scheme and YCK-IBS scheme exhibit the fast speed in the **Sign** phase. Since the proposed IBS scheme only needs one pairing computation, it provides much faster **Verify** than other IBS schemes.

- **Case 4:** When the elliptic curve is $E_{ns4}^{2048}(\mathbb{F}_{512})$, the embedding degree is 4 and the timing for computing exponentiations on G_3 cannot be ignored. From the results shown in Table 4.5, **Verify** of the proposed IBS scheme is still the fastest one while **Sign** does not exhibit advantage in comparison with other IBS schemes. It is noted that Yi-IBS scheme and YCK-IBS scheme are efficient in the **Sign** phase because they do not need to compute exponentiation on G_3 . However, their performance are at most equivalent to the proposed IBS scheme. the proposed IBS scheme provides the fastest **Sign** and **Verify**.

- **Case 5:** When the elliptic curve is $E_{ns8}^{4096}(\mathbb{F}_{512})$, it is observed that the timing for computing exponentiation on G_3 is very time consuming because the embedding degree is 8. **Sign** of the proposed IBS scheme is the

Table 4.7: Timings of the proposed schemes and other IBS schemes on $E_{ns8}^{4096}(\mathbb{F}_{512})$

Schemes	Precomputation (<i>ms</i>)	Setup (<i>ms</i>)	Extract (<i>ms</i>)	Sign (<i>ms</i>)	Verify (<i>ms</i>)
the proposed IBS scheme	361.07	6.61	6.63	63.22	417.7
Hess-IBS	361.07	6.61	19.96	63.22	792.10
Yi-IBS	N/A	6.61	19.96	19.83	742.10
YCK-IBS	N/A	6.61	19.96	33.18	742.10

slowest one when compared to other schemes. Although research work in [96] provides a fast algorithm for computing exponentiation such that computing a Tate pairing is more efficient than computing a Weil pairing at high security levels, their method cannot reduce the running time of the **Sign** phase, because there is only an exponentiation computation on G_3 . However, their method does reduce the timing of the **Verify** phase. The results shown in Table 4.7 are not based on the method of [96], but this does not influence the comparison. Table 4.7 shows that the proposed IBS scheme in the random oracle model still has the fastest **Verify**. The fastest **Sign** and **Verify** are provided by Yi-IBS scheme and the proposed IBS scheme.

4.6 Summary

In this chapter, the proposed IBS scheme, an efficient IBS scheme using the random oracles, is first introduced. Its security depends on a variant of k -CAA. This new IBS scheme in the random oracle model improves the efficiency of extracting secret keys and verifying signatures by eliminating the special hash

function called the *MapToPoint* function. The experimental results shown in Table 4.3-4.7 indicate that this new IBS scheme is the most efficient one for key extraction and verification compared to all other existing IBS schemes based on the bilinear pairings at the security levels of 1024 bits and 2048 bits. In particular, the efficiency of verification is improved by at least 50% when a supersingular elliptic curve with an embedding degree 2 is used. A simple method of building suitable non-supersingular elliptic curves is presented and two examples are given.

Chapter 5

Applications of Identity-Based Signature Schemes

In Chapter 4, we introduced two new IBS schemes in the random oracle model. It has been proven to be secure in the random oracle model. The investigation of the performance of the proposed IBS scheme showed that it is more efficient than other IBS schemes in the random oracle model that use bilinear pairings at the 1024 bits and 2048 bits security levels. In this chapter, some applications based on the proposed IBS scheme will be presented.

5.1 Introduction

In traditional Public Key Cryptography, digital signature is a necessary and basic cryptographic primitive as it provides authentication, integrity and non-repudiation. With the growth of electronic commerce and the popularization of the Internet, traditional digital signatures (such as RSA signatures and DSS)

were unable to satisfy the requirements of practical applications in terms of efficiency and features. Researchers have developed many solutions to improve the efficiency of digital signatures to extend their functions. In the recent years, ID-PKC has made great progress since the bilinear pairings were discovered to construct positive cryptographic schemes. This discovery spurred researchers to propose various identity-based cryptographic schemes. Today, designing efficient identity-based cryptographic schemes is one of the most active research areas in the community. Another active area is constructing new cryptographic schemes by combining traditional cryptographic schemes with identity-based cryptographic schemes. These include identity-based blind signatures (IBBS) [178], identity-based ring signatures (IBRS) [178] and identity-based chameleon signatures (IBCS) [9].

In this chapter, we concentrate on applications based on the IBS scheme in the random oracle model proposed in Chapter 3. These applications include batch verification on the proposed IBS scheme, IBBS scheme and IBCS scheme. Batch verification is used to reduce the time taken to verify signatures when the verifier needs to verify a large number of signatures. An IBBS scheme is the combination of an IBS scheme and a blind signature scheme [56]. It provides the anonymity property like a blind signature scheme, while simplifies key management at the same time. Similarly, an IBCS scheme is the combination of an IBS scheme and a chameleon hash function [110]. It offers non-transferability like a chameleon signature scheme, while also provides the advantage of simplifying key management.

5.1.1 Contributions

The first contribution is the proposal to carry out batch verification of the proposed IBS scheme in the random oracle model. Batch verification is a useful method when many signatures have to be verified. Using this method, the verifier has an ability to verify multiple signatures simultaneously so as to reduce the time taken to verify multiple signatures. According to Yoon et al.'s taxonomy [175], batch verification can be classified into three types:

- **Type 1:** Multiple signers sign a single message to obtain multiple signatures.
- **Type 2:** A single signer signs multiple messages to obtain multiple signatures.
- **Type 3:** Multiple signers sign multiple messages to obtain multiple signatures. Note that all the messages are distinct, and so are the signers.

Most existing IBS schemes support the first two types. Only YCK-IBS scheme [175], supports the third type. In the first part of this chapter, we will investigate how the proposed IBS scheme supports Type 2 and 3 batch verification. The results illustrate that the proposed IBS scheme performs better for carrying out Type 3 batch verification compared to YCK-IBS scheme.

The second contribution is to propose a new IBBS scheme based on the proposed IBS scheme. An IBBS scheme is the combination of a blind signature scheme and an IBS scheme. It provides the same properties as blind signatures together with the advantage of simplifying the key distribution due to the use of the user identity as the public key. Similar to blind signature [56], IBBS

requires an interactive protocol between the signer and the user. During the process of generating blind signatures, the user knows the message for signing but does not know the key to be used for signing, while the signer knows the key for signing but does not know the message to be signed. Due to this feature, blind signatures and IBBS are thought to be very useful tools in anonymous environments (such as electronic cash, electronic bidding and anonymous electronic voting systems) and have been deployed in many real applications [83, 154, 105]. Relying on the proposed IBS scheme described in Chapter 4, we develop a new IBBS scheme which extracts the secret key faster and has a more efficient blind signature issuing protocol than other IBBS schemes. Furthermore, the new IBBS scheme requires less space for the storage of public parameters in the blind signature issuing protocol than other IBBS schemes.

The third contribution is to develop a new IBCS scheme. A chameleon signature scheme [110] is a non-interactive signature based on a special hash function, called a chameleon hash function [110]. It provides non-transferability; in other words, after the designated recipient receives a signature, he cannot display the correctness of this signature to other parties. An IBCS scheme is the combination of a chameleon hash function and an IBS scheme. Thus, it not only provides non-transferability but also simplifies key management due to using the user identity as the public key. Zhang et al. [181] described two identity-based chameleon hash functions (ZSS-IBCHF-I and ZSS-IBCHF-II) based on the bilinear pairings and combined them with IBS schemes that used bilinear pairings to construct IBCS schemes. Regrettably, IBCS scheme based on ZSS-IBCH-II requires **TA** (Trusted Authority) to provide two different key extraction algorithms so that two different public/secret key pairs for the IBCH and IBS system can be generated. It is a very inefficient method. The new

IBCS addresses this problem. We construct a new IBCS where the proposed IBS scheme in the random oracle model shares the same key extraction algorithm as ZSS-IBCHF-II [181].

5.1.2 Prior Research

A Brief Survey of Batch Verification: After the introduction of digital signatures into cryptography, researchers began considering how the processes of signing messages and verifying signatures could be carried out more efficiently. Batch verification, which has been in development for many years, is one of the tools that has been frequently evaluated for these purposes, as it helps shorten the time needed for the verification of multiple signatures. In the last two decades, various researchers have discussed batch verification [27, 134, 173, 132, 84], and we summarize their contributions below.

The multisignature proposed by Itakura and Nakamura [116] was the earliest batch verification method available for a signature scheme. It allowed any subgroup of a group of users to jointly sign a message such that a verifier would be convinced that each member of the subgroup participated in the signing. According to the classification described in Section 5.1.1, this method belongs to the first type of batch verification. Multisignature schemes based on RSA and DLP were proposed in [116, 139, 140, 126]. Following that, Boldyreva [30] proposed the first multisignature scheme using bilinear pairings.

The key feature of the multisignature method is that all the signatures to be verified have been generated on the same message. Type 2 and 3 batch verification methods, described in Section 5.1.1, offer more flexible mechanisms. Specifically, these two methods can be applied to verify multiple signatures on

different messages. The difference between the Type 2 and 3 methods is that while the multiple signatures that are verified by using the Type 2 method are generated on multiple messages by a single signer, the multiple signatures verified with the Type 3 method are generated on multiple messages by multiple signers.

The condensed RSA method proposed by Mykletun et al. [133] is classified as a Type 2 method. It compresses multiple RSA signatures on different messages generated by one signer into one signature. Conversely, the aggregated signatures [159, 38, 39] belong to Type 3. There are two types of aggregate signatures: general aggregate signatures and sequential aggregate signatures. In a general aggregate signature scheme, each user U_i signs a message m_i to obtain a signature σ_i . Anyone who knows the public aggregation algorithm can compress multiple signatures into a single signature. Signature aggregation is completed after the messages have been signed. But in a sequential aggregate signature scheme, signature aggregation must be carried out during the signing of the messages.

Besides aggregate signatures, Bellare, Garay and Rabin [25] developed a special Type 3 batch verification. They defined a new but weak variant of batch verification, called *screening*, when they made a systemic analysis of batch verification for modular exponentiation and digital signatures from RSA. The aim of screening is to determine whether the signer has at some point authenticated the text, rather than to check that the particular string provided is a valid signature of the original text. Yoon et al. [175] proposed an IBS scheme (YCK-IBS scheme) which supported all types of batch verification. Their analysis also showed that both Hess-IBS [100] and CC-IBS [54] supported Type 1 and 2

batch verification methods but not Type 3.

A Brief Survey of Identity-Based Blind Signature: The concept of blind signatures was introduced by Chaum [56] in 1983. Blind signatures provide anonymity and have been deployed in real applications, such as electronic cash, electronic bidding, electronic voting systems. Researchers have proposed a range of secure blind signature schemes and developed some variants [147, 106, 148, 24, 30, 45]. Some of these blind signature schemes [147, 148, 24, 30] have been proven to be secure in the random oracle model while others [106, 45] have been proven to be secure in the standard model.

The earliest blind signature scheme proposed by Chaum [56] was based on RSA. However, its security was not proven until Bellare et al. [24] introduced the computational assumption of “chosen-target-one-more RSA-inversion”. Subsequently, Pointcheval and Stern [147] presented a blind Schnorr signature scheme. The first version of blind signatures using bilinear pairings was proposed by Boldyreva [30]. Its security depends on the chosen-target computational Diffie-Hellman problem [30]. Following that, Zhang and Kim [178] introduced the first identity-based blind signature (IBBS) scheme in 2002. To distinguish this scheme from another IBBS developed by Zhang and Kim [179], we denote this scheme as ZK-IBBS-I scheme in this thesis. The security of ZK-IBBS-I scheme depends on a complexity assumption called the ROS-problem [156]. However, Wagner [170] put forward the claim that the ROS-problem could be broken in sub-exponential time. Thus, ZK-IBBS-I scheme had to increase its related system parameters to resist such an attack. Unfortunately, this resulted in a reduction of its efficiency. To overcome this problem, Zhang and Kim [179] proposed another IBBS (ZK-IBBS-II) scheme. ZK-IBBS-II scheme did not de-

pend on the ROS-problem and provided more efficient performance compared to ZK-IBBS-I scheme.

The basic requirement of a blind signature scheme is that the message to be signed must be known only to the user, and remain unknown to the signer. However, certain applications require that part of the message to be signed be publicly observable to the signer. Abe and Fujisaki [2] proposed the concept of the partially blind signature to address this issue. Their partial blind signature scheme is based on RSA and requires common information to be shared between the user and the signer. Abe and Camenisch [1] also describe a partial blind signature scheme based on the Schnorr signature [155]. Further, Miyazaki et al. [131] presented a partial blind signature scheme based on DSS and also constructed a blind signature scheme with message recovery. Fan and Lei [82] proposed an efficient partial blind signature scheme which only required several modular additions and modular multiplications. While Chien et al. [66] proposed an RSA version of partial blind signatures, Abe and Okamoto [3] provided the first provably secure partial blind signature scheme based on DLP. Following on from a short signature scheme proposed in [182], Zhang et al. [183] developed an identity-based partial blind signature scheme. However, Chow et al. [68] pointed out that Zhang et al.'s identity-based partial blind signature scheme was insecure. They thus proposed a more secure identity-based partial blind signature scheme and a certificate-based partial blind signature scheme.

A Brief Survey of Identity-based Chameleon Signatures: Chameleon signatures, introduced by Krawczyk and Rabin [110], are based on a special hash function called the chameleon hash function. The chameleon hash function is a trapdoor one-way hash function. Without the knowledge of the associated trapdoor, the

chameleon hash function is resistant to the computation of pre-images and of collisions. In contrast, with the knowledge of the trapdoor, anyone can easily compute the collisions. Before chameleon signatures were introduced into cryptography, only undeniable signatures [59, 57, 42, 60, 101, 102] provided the properties of non-repudiation and non-transferability. However, undeniable signatures usually depended on an interactive protocol, resulting in inefficient performance. Although some non-interactive versions of undeniable signatures [102] have been proposed, their performance has not been satisfactory. Because chameleon signatures offer non-repudiation and non-transferability as well as the efficiency of regular digital signatures, chameleon signatures are seen as an alternative of undeniable signatures.

The first chameleon hash function was introduced by Krawczyk and Rabin [110]. Its security depended on the classic cryptographic problems (IFP and DLP). Based on this chameleon hash function, Krawczyk and Rabin [110] proposed the first chameleon signature scheme whose security depended on the security of regular digital signatures (such as RSA or DSS). In 2004, Ateniese and Medeiros [9] first constructed an identity-based chameleon hash function (IBCHF), and presented an IBCS scheme by combining their IBCHF with IBS schemes. Because their IBCHF was constructed based on RSA, the security of their IBCS depended on both secure digital signatures and IBS schemes such as [161, 98]. Zhang et al. [181] proposed two different IBCHF based on the bilinear pairings and constructed IBCS schemes by combining their IBCHF functions with IBS schemes based on the bilinear pairings. Recent work on chameleon signature scheme is by Chen et al. [64] who used “customized identities” [9] to solve the problem of key exposure which existed in the original chameleon signature scheme.

The rest of this chapter is organized as follows: Section 5.2 introduces Type 3 batch verification of the proposed IBS scheme in the random oracle model and analyzes its security. Section 5.3 introduces a new IBBS scheme by using the proposed IBS scheme in the random oracle model and the related security analysis. Section 5.4 describes how an identity-based chameleon signature scheme based on the proposed IBS scheme in the random oracle model which has better performance. Finally, Section 5.5 summarizes the chapter.

5.2 Batch verification

5.2.1 Definition

The aim of batch verification is to verify multiple signatures simultaneously such that the time for verification is significantly shortened. Its definition is as follows [175]:

Definition 5.1. *Given multiple signatures $\sigma_1, \sigma_2, \dots, \sigma_n$ on the messages $m_1, m_2, \dots, m_n$ and the corresponding identities $id_1, id_2, \dots, id_n$, a verifier checks the validity of all (or some of) the signatures at once.*

Yoon et al. [175] formalized the notion of the attack model of batch verification of Type X ($X = 1, 2, 3$) on the IBS scheme. We say that $\mathcal{F}$ is a λ -batch forger of Type X on IBS schemes if it wins the following game [175]:

- **Setup:** $\mathcal{F}$ is given public parameters.
- **Queries:** $\mathcal{F}$ issues the queries to the hash, extract and sign oracle by his choices and obtains the hash values of his chosen messages, the secret

keys of his chosen identities and the corresponding signatures by his chosen identities and messages.

- **Outputs:** Finally, $\mathcal{F}$ outputs an integer n which is not larger than λ , $id_1, id_2, \dots, id_n$ and messages $m_1, m_2, \dots, m_n$ and the corresponding signatures $\sigma_1, \sigma_2, \dots, \sigma_n$ of Type X . Note that id_n must not be queried by the extract oracle, and (id_n, m_n) must not be queried by the sign oracle. $\mathcal{F}$ wins the game if $\mathcal{F}$'s outputs pass the batch verification successfully.

In the above game, $\mathcal{F}$ is given the power to access the extract oracle to obtain all the users' secret keys except id_n and access the sign oracle on all the messages except m_n . From the above description, the definition of secure batch verification on IBS schemes is as follows [175]:

Definition 5.2. *In the random oracle model, a λ -batch forger $\mathcal{F}(t, q_H, q_E, q_S, \lambda, \epsilon)$ breaks the batch verification on some IBS scheme by the adaptive chosen message and identity attack if $\mathcal{F}$ runs in time at most t , makes at most q_H queries to the hash oracle, and q_E extract queries and q_S signature queries with the probability of at least ϵ to generate at most λ signatures which successfully pass the batch verification.*

Remark 5.1. *The above definition provides only screening which is a special batch verification proposed by Bellare et al. [25]. Signature screening provides a “weak but fast” method for verifying signatures compared with verifying signatures one by one. Thus, the batch verification of the proposed IBS scheme introduced in the following section provides only screening.*

5.2.2 Batch Verification On The Proposed IBS Scheme

As described in [175], both Hess-IBS scheme [100] and CC-IBS scheme [54] supported batch verification of Type 2. However, only the IBS scheme in [175] supported batch verification of Type 3. This section investigates how the proposed IBS scheme in the random oracle model supports both Type 2 and Type 3 batch verification. The following depicts the implementation of Type 2 and 3 batch verification on the proposed IBS scheme. The public parameters $(p, G_1, G_3, \hat{e}_t, P_1, P_{pub}, \omega, H_1, H_2)$ have been introduced in Section 4.3.2.

- **Batch Verification for Type 2:** Suppose a signer with identity id generates the signatures $(r_1, V_1), (r_2, V_2), \dots, (r_\lambda, V_\lambda)$ on at most λ distinct messages $m_1, m_1, \dots, m_\lambda$. Then, a verifier can verify these signatures simultaneously using the following:

1. $u_i = H_2(m_i, r_i)$
2. $\omega^{\sum_{i=1}^{\lambda} u_i} \prod_{i=1}^{\lambda} r_i = \hat{e}_t(P_{pub} + H_1(id) \cdot P_1, \sum_{i=1}^{\lambda} V_i)$

- **Batch Verification for Type 3:** Suppose there are at most λ signatures $(id_1, m_1, r_1, V_1), (id_2, m_2, r_2, V_2), \dots, (id_\lambda, m_\lambda, r_\lambda, V_\lambda)$ where all the messages and identities are distinct. Then, the verifier can verify these signatures simultaneously by using the following:

1. $u_i = H_2(m_i, r_i),$
2. $\omega^{\sum_{i=1}^{\lambda} u_i} \prod_{i=1}^{\lambda} r_i = \hat{e}_t(P_{pub}, \sum_{i=1}^{\lambda} V_i) \hat{e}_t(P_1, \sum_{i=1}^{\lambda} H_1(id_i) \cdot V_i)$

In the next section, we focus on the proof of the security of Type 3 batch verification on the proposed IBS scheme. The proof of the security of batch verification of Type 2 on the proposed IBS scheme is similar.

5.2.3 The Security of Type 3 Batch Verification

The security of Type 3 batch verification of the proposed IBS scheme in the random oracle depends on the following theorem:

Theorem 5.1. *In the random oracle model, if a λ -batch forger $\mathcal{F}$ $(t, q_H, q_E, q_S, \lambda, \epsilon)$ -breaks the Type 3 batch verification of the proposed IBS scheme in the random oracle model by using an adaptive chosen message and identity attack, then another (t', ϵ') -algorithm $\mathcal{C}$ exists which is able to solve Generalized k -CAA, where $t' = t$, $q_S \leq q_H$, $k = q_S$ and $\frac{(p-q_S) \cdot q_E^{q_S} \cdot q_S^{q_S}}{p^{q_E+1} \cdot q_{H_1}^{q_E} \cdot q_H^{q_S}} \cdot \epsilon$.*

Proof. Suppose the algorithm $\mathcal{F}$ is a λ -batch forger that $(t, q_H, q_E, q_S, \lambda, \epsilon)$ -breaks the proposed IBS scheme. We wish to construct another algorithm $\mathcal{C}$ to solve Generalized k -CAA. In the following game, $\mathcal{C}$ plays the role of challenger and interacts with the forger $\mathcal{F}$:

- **Setup:** Algorithm $\mathcal{C}$ runs **Setup** and sends $\mathcal{F}$ the public parameters $(p, G_1, G_3, \hat{e}_t, P_1, P_{pub}, \omega, H_1, H_2)$, where $P_{pub} = xP_1$ and x is a random element in $\mathbb{Z}_p^*$.
- **Queries:** $\mathcal{F}$ makes the following queries (without loss of generality, we assume that, for any key extract query or signature query involving an identity, an identity query was previously issued for the same identity):
 1. *Preparing query:* $\mathcal{C}$ prepares a response set $\{d_1, d_2, \dots, d_{H_1}\}$ of the identity hash oracle queries on id_i where $d_i \in \mathbb{Z}_p^*$. The response set includes a subset $\{e_1, e_2, \dots, e_{q_E}\}$. Each $e_{i'}$ of the subset is distributed randomly in the response set.

2. *Identity hash query:* $\mathcal{F}$ makes a hash oracle query on id_i for $1 \leq i \leq q_{H_1}$, $\mathcal{C}$ sends d_i to $\mathcal{F}$ as the response of the hash query on id_i .
3. *Key extract query:* $\mathcal{F}$ makes an extract query on d_i (where d_i represents the identity id_i) for $1 \leq i \leq q_E$. if $d_i = e_{i'}$, $\mathcal{C}$ responds $S_{id_i} = \frac{1}{x+e_{i'}}P_1$ as the response (Note that x is a unknown value, computing $S_{id_i} = \frac{1}{x+e_{i'}}P_1$ is based on some probability of computing the value of S_{id_1} which is analyzed in the discussion below). Otherwise, $\mathcal{C}$ stops the simulation and report failure.
4. *Message Hash Query:* $\mathcal{C}$ constructs a H-list of tuple $\langle m_i, r_i, u_i, s_i \rangle$ ($1 \leq i \leq q_H$) for responding to $\mathcal{F}$'s queries on *the message hash query*. When adversary $\mathcal{F}$ queries the hash oracle on message m_i , the H-list is changed as follows: If $\mathcal{F}$ sends a query for message m_i which has appeared in H-list, then $\mathcal{C}$ answers with the corresponding (r_i, u_i, s_i) to $\mathcal{F}$. Otherwise, $\mathcal{C}$ picks a random element $s_i \in \mathbb{Z}_p^*$ and a random element $u_i \in \mathbb{Z}_p^*$, and then computes $r_i = w^{s_i}$. Let $u_i = H_2(m_i, r_i)$ such that $s_i + u_i \neq s_j + u_j$ when $i \neq j$. Each $\langle m_i, r_i, u_i, s_i \rangle$ is added into the H-list. In addition, $\mathcal{C}$ maintains another set $S = \{h_1, h_2, \dots, h_{q_H}\}$ where $h_i = u_i + s_i$.
5. *Sign Query:* For any given identity-message pair (id_i, m_j) , $\mathcal{C}$ responds to $\mathcal{F}$'s queries on the sign oracle as follows: $\mathcal{C}$ scans the H-list to check whether m_j is in the list or not. If not, $\mathcal{F}$ stops the simulation and reports failure. Otherwise, $\mathcal{F}$ obtains the corresponding r_j, u_j, s_j . Since $\mathcal{F}$ is a Type 3 λ -batch forger that requires multiple signatures on multiple messages generated by multiple signers, a distinct message must be signed by a distinct user. There is a one-to-one map relationship between the user set $U: \{id_1, id_2, \dots, id_{q_E}\}$

and the message set M : $\{m_1, m_2, \dots, m_{q_S}\}$. We might as well think that the signature on the message m_i for the user id_j is discarded if $i \neq j$. Suppose $\mathcal{C}$ computes $\delta_j = u_j + s_j$ such that $\delta_j \in \{h_1, h_2, \dots, h_{q_H}\}$ ($q_S \leq q_H$), and then computes the signature $V_j = \delta_j S_{id_j}$. Otherwise, $\mathcal{C}$ stops the simulation and reports failure. Finally, r_j and V_j are sent to $\mathcal{F}$ as the response of the *sign query*.

- **Output:** Eventually, $\mathcal{F}$ stops the simulation and returns the following values: a value n , n identities $id_1, id_2, \dots, id_n$, n messages $m_1, m_2, \dots, m_n$ and n signatures $(r_1, V_1), (r_2, V_2), \dots, (r_n, V_n)$. Notes that id_n and m_n must not be queried by the extract oracle and the sign oracle, respectively. The corresponding H-list is $\langle m_i, r_i, u_i, s_i \rangle$ where $1 \leq i \leq (n-1)$. $\mathcal{F}$ wins the game only if the following conditions are satisfied:

1. $\mathcal{F}$'s outputs pass the batch verification,
2. There is a one-to-one map between the user set U and the message set M . The distinct message must be signed for the distinct user.

Suppose $r_n = \omega^{s_n}$, let $u_n = H_2(m_n, r_n)$, where s_n and u_n are randomly chosen in $\mathbb{Z}_p^*$ such that $\delta_n = u_n + s_n \notin \{h_1, h_2, \dots, h_{q_H}\}$. Since $\mathcal{F}$'s outputs are $(id_1, m_1, r_1, V_1), (id_2, m_2, r_2, V_2), \dots, (id_n, m_n, r_n, V_n)$ pass the batch verification. Then

$$\omega^{\sum_{i=1}^n u_i} \prod_{i=1}^n r_i = \hat{e}_t(P_{pub}, \sum_{i=1}^{n-1} V_i + V_n) \hat{e}_t(P, \sum_{i=1}^{n-1} id_i \cdot V_i + H_1(id_n) \cdot V_n) \quad (5.1)$$

In addition, $(id_1, m_1, r_1, V_1), (id_2, m_2, r_2, V_2), \dots, (id_{n-1}, m_{n-1}, r_{n-1}, V_{n-1})$ must pass the batch verification. Therefore, the following formula

is correct:

$$\omega^{\sum_{i=1}^{n-1} u_i} \prod_{i=1}^{n-1} r_i = \hat{e}_t(P_{pub}, \sum_{i=1}^{n-1} V_i) \hat{e}_t(P, \sum_{i=1}^{n-1} H_1(id_i) \cdot V_i) \quad (5.2)$$

Since $\omega = \hat{e}_t(P, P)$ and $r_i = w^{s_i}$, combining Equations 5.1 and 5.2, we have:

$$\hat{e}_t(P, P) = \hat{e}_t((x + H_1(id_n))P, V_n)^{1/(u_n + s_n)}$$

Hence, $V_n = \frac{u_n + s_n}{x + H_1(id_n)} P = \frac{\delta_n}{s + H_1(id_n)} P$. Since $\delta_n \notin \{h_1, h_2, \dots, h_{q_H}\}$ and id_n is not queried by the extract oracle, $\mathcal{C}$ outputs V_n as the solution of Generalized k -CAA. (Actually, V_n is the solution of a special instance of Generalized k -CAA: given a tuple $(f_1, f_2, \dots, f_k, g_1, g_2, \dots, g_k \in \mathbb{Z}_p^*, P \in G_1, xP, \frac{f_1}{x+g_1}P, \frac{f_2}{x+g_2}P, \dots, \frac{f_k}{x+g_k}P)$, where $f, g \in \mathbb{Z}_p^*, f \notin \{f_1, f_2, \dots, f_k\}$ and $g \notin \{g_1, g_2, \dots, g_k\}$, compute $\frac{f}{x+g}P$).

In the *Identity query* phase, $\mathcal{C}$ stops the simulation and reports failure when $d_i \neq e_{i'}$. The probability of this event is that $\mathcal{C}$ successfully computes S_{id_i} is $\frac{q_E}{q_{H_1}}$. The probability that $\mathcal{C}$ successfully computes S_{id_i} is $\frac{q_E}{p \cdot q_{H_1}}$. For all identity oracle queries, $\mathcal{C}$ will not fail with the probability $(\frac{q_E}{p \cdot q_{H_1}})^{q_E}$. In the *Sign query* phase, $\mathcal{C}$ aborts the simulation only when $\delta_i \notin \{h_1, h_2, \dots, h_{q_H}\}$. The probability that $\mathcal{F}$'s outputs pass batch verification is at least q_S/q_H . Thus, for all sign queries, the probability that $\mathcal{C}$'s outputs pass batch verification is at least $(q_S/q_H)^{q_S}$. The probability of another event, $\delta_n \notin \{h_1, h_2, \dots, h_{q_H}\}$, is $1 - \frac{q_S}{l}$. The probability that $\mathcal{C}$ successfully outputs the solution of k -CAA is $\frac{(p-q_S) \cdot q_E^{q_E} \cdot q_S^{q_S}}{p^{q_E+1} \cdot q_{H_1}^{q_E} \cdot q_H^{q_S}} \cdot \epsilon$. $\mathcal{C}$'s running time is identical to $\mathcal{F}$'s running time, $t = t'$. $\square$

5.2.4 Performance

This section will investigate the performance of Type 3 batch verification of the proposed IBS scheme in the random oracle model and compare it with YCK-IBS scheme.

Table 5.1: The comparison of batch verification on the proposed scheme and other IBS schemes

Verify n signatures by different methods			
n	Batch verification on the proposed IBS scheme	Verification using the proposed scheme by the traditional method	Batch verification on YCK-IBS scheme
5	$2P_a + 1E_{G_3} + 4M_{G_3} + 5M_{G_1}$	$5(P_a + E_{G_3} + M_{G_3} + M_{G_1})$	$6P_a + 5(M_{G_1} + H_{M_{G_1}})$
10	$2P_a + 1E_{G_3} + 9M_{G_3} + 10M_{G_1}$	$10(P_a + E_{G_3} + M_{G_3} + M_{G_1})$	$11P_a + 10(M_{G_1} + H_{M_{G_1}})$
50	$2P_a + 1E_{G_3} + 49M_{G_3} + 50M_{G_1}$	$50(P_a + E_{G_3} + M_{G_3} + M_{G_1})$	$51P_a + 50(M_{G_1} + H_{M_{G_1}})$
100	$2P_a + 1E_{G_3} + 99M_{G_3} + 100M_{G_1}$	$100(P_a + E_{G_3} + M_{G_3} + M_{G_1})$	$101P_a + 100(M_{G_1} + H_{M_{G_1}})$
200	$2P_a + 1E_{G_3} + 199M_{G_3} + 200M_{G_1}$	$200(P_a + E_{G_3} + M_{G_3} + M_{G_1})$	$201P_a + 200(M_{G_1} + H_{M_{G_1}})$
500	$2P_a + 1E_{G_3} + 499M_{G_3} + 500M_{G_1}$	$500(P_a + E_{G_3} + M_{G_3} + M_{G_1})$	$501P_a + 500(M_{G_1} + H_{M_{G_1}})$

Table 5.2: Timings of batch verification of the proposed IBS scheme and other IBS schemes on $E_{ss2}^{1024}(\mathbb{F}_{512})$

	Batch verification on the proposed IBS scheme	Verification using the proposed IBS scheme by the traditional method		Batch verification on YCK-IBS scheme	
n	Time (ms)	Time (ms)	Impro.	Time (ms)	Impro.
5	67.57	136.1	50.3%	228.98	70.5%
10	92.67	272.2	66.0%	437.68	78.8%
50	293.47	1361	78.4%	2107.28	86.1%
100	544.47	2722	80.0%	4194.28	87%
200	1046.47	5444	80.8%	8368.28	87.5%
500	2552.47	13610	81.2%	20890.28	87.8%

To verify n signatures on n distinct messages for n distinct signers without

using the batch verification method, the proposed IBS scheme requires $nM_{G_1} + nE + nM_{G_3} + nPa$. Type 3 batch verification based on YCK-IBS scheme requires $(n + 1)Pa + nM_{G_1} + nH_{M_{G_1}}$. However, using batch verification on the proposed IBS scheme only requires $2Pa + 1E + (n - 1)M_{G_3} + nM_{G_1}$. The comparison is shown in Table 5.1. From the results shown in Table 4.1 of Chapter 4, in case of $E_{ss2}^{1024}(\mathbb{F}_{512})$, batch verification of YCK-IBS scheme requires about $(41.74n + 20.28)ms$. While the proposed IBS scheme requires about $(27.22n)ms$ for verification without using the batch verification method, using batch verification reduces the time requires to about $(5.02n + 42.47)ms$. As shown in Table 5.2, when n is a large number (e.g. $n \geq 100$), using batch verification on the proposed IBS scheme significantly improves the performance of signature verification, the efficiency can be improved at least 80%.

The reason of fast verifying signature is the time of computing pairings. Table 4.2 shows that the computation of pairings takes up much longer time than other cryptographic primitives. While the proposed IBS scheme only requires one pairing computation, YCK-IBS scheme needs to compute pairings twice and thus takes additional time to verify signatures. The timing of Table 5.2 show that the proposed IBS scheme provides faster verification on many signatures even when batch verification is not used. Table 5.2 shows clearly that using batch verification on the proposed IBS scheme reduces the time required to verify multiple signatures. Verifying 100 signatures by batch verification saves 80% time taken as compared to verifying signatures one at a time.

5.3 A New Identity-Based Blind Signature Scheme

5.3.1 Definition

Blind signatures are very useful in anonymous environments such as anonymous electronic cash or electronic voting systems. ID-PKC is an alternative to certificate-based Public Key Cryptography, and an IBBS schemes are the combination of a blind signature scheme and an IBS scheme. Without changing the property of blind signatures, IBBS scheme provides the advantage of simplifying key management. The formal definition of IBBS scheme is given by Zhang and Kim [178]:

Definition 5.3. *An IBBS scheme includes a trusted third party: the Trusted Authority (**TA**), and five algorithms: **Setup**, **User**, **Extract**, **Signer**, **Verify**, where*

- *As the trusted third party, **TA** is a trustee who provides a secure channel to transfer secret information to users. It runs **Setup** to generate public parameters and the master key, and runs **Extract** to extract secret keys for users.*
- **Setup:** *This is a probabilistic polynomial algorithm which takes as input a security parameter κ and returns the related parameters and the master key.*
- **Extract:** *This is a probabilistic polynomial algorithm which takes as its input public parameters, the master key and an arbitrary $id \in \{0,1\}^*$ or $\mathbb{Z}_p^*$ (where id is a signer's identity and works as the signer's public key), and return a secret key S_{id} .*

- **Signer and User:** These are a pair of probabilistic algorithms. They are seen as two interactive Turing machines where both machines have the following tapes: a read-only input tape, a write-only output tape, a read-/write working tape, a read-only random tape, and two communication tapes. **Signer** is provided on the input tape of the machine (id, S_{id}) , as is **User** (id, m) , where m is a message to be signed. The length of all inputs must be polynomial in κ (where κ is the security parameter). **Signer** and **User** engage in the signature issuing protocol and stop in polynomial time. At the end of this protocol, **Signer** outputs either 'completed' or 'not-completed', while **User** outputs either 'fail' or a valid signature σ of the message m .
- **Verify:** This is a probabilistic polynomial-time algorithm which takes as its input public parameters and (id, m, σ) . If the candidate signature is valid, it outputs true. Otherwise, it outputs false.

A secure IBBS scheme must provide two properties: blindness and unforgeability. The requirement for the unforgeability property is the same as the regular signature scheme and IBS scheme. The following shows the requirement of the blindness property [106, 178]:

Definition 5.4. Let $\mathcal{A}$ denote a probabilistic polynomial-time algorithm which controls **Signer**. In the following game, $\mathcal{A}$ will interact with two honest users: U_0 and U_1 :

1. Given the public parameters and the identity id , **TA** extracts the corresponding secret key S_{id} ;
2. Given the identity id and the corresponding secret key S_{id} , $\mathcal{A}$ generates

- two independent messages: m_0 and m_1 ;
3. Pick a random bit b from $\{0, 1\}$ which is secret for $\mathcal{A}$. Place m_b and m_{1-b} on the read-only input tape of U_0 and U_1 , respectively;
 4. $\mathcal{A}$ joins the blind signature issuing protocol with U_0 and U_1 in arbitrary order;
 5. Suppose that the outputs of U_0 and U_1 are σ_b on the message m_b and σ_{1-b} on the message m_{1-b} under their private tapes, respectively. Then, offer those outputs to $\mathcal{A}$. Otherwise, send $\perp$ to $\mathcal{A}$;
 6. $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$.

If $b' = b$, $\mathcal{A}$ knows the message and its corresponding signature of each user. In this case, we say $\mathcal{A}$ wins the above game.

An IBBS scheme is blind if a probabilistic polynomial-time algorithm $\mathcal{A}$ wins the above game with the probability of at most $1/2 + 1/\kappa^c$ for a sufficiently large κ and some constant c . The probability is taken over the coin flips of **Extract**, two users, and $\mathcal{A}$.

5.3.2 The Proposed IBBS Scheme

Figure 5.1 and 5.2 shows the blind signature issuing protocol of ZK-IBBS-I and ZK-IBBS-I II schemes respectively. Figure 5.3 shows the blind signature issuing protocol of the proposed IBBS scheme. The details of the proposed IBBS scheme are described as follows:

- **Setup:** Suppose that H_1 and H_3 are two collision-resistant hash functions. H_1 is defined as $\{0, 1\}^* \rightarrow \mathbb{Z}_p^*$ and H_3 is defined as $\{0, 1\}^* \times G_1^* \rightarrow \mathbb{Z}_p^*$.

Figure 5.1: The blind signature issuing protocol of ZK-IBBS-I scheme

Figure 5.2: The blind signature issuing protocol of ZK-IBBS-II scheme

^aThe hash function H_3 is defined as $\{0, 1\} \times G_1^* \rightarrow \mathbb{Z}_p^*$.

Figure 5.3: The blind signature issuing protocol of the proposed IBBS scheme

Given the security parameter $\kappa \in \mathbb{N}$, **TA** chooses randomly $x \in \mathbb{Z}_p^*$ and $P_1 \in G_1$, and publishes $\{p, G_1, G_3, \hat{e}_t, H_1, H_3, P_1, P_{pub}\}$ as the public parameters, where $P_{pub} = xP$. x is the master key and kept secret.

- **Extract:** Given an identity id , **TA** computes the corresponding secret key $S_{id} = \frac{1}{x+H_1(id)}P_1$.
- **Blind signature issuing protocol:**
 - The signer randomly chooses an element r from $\mathbb{Z}_p^*$, and computes $U = rP_1$, then sends U to the user as a commitment.
 - **User (Blinding):** The user randomly chooses α, β from $\mathbb{Z}_p^*$ as blinding factors, and computes $U' = \alpha U + \alpha\beta P_1$, $h = \alpha^{-1}H_3(m, U') + \beta \pmod{p}$, then sends h to the signer.
 - **Signer (Signing):** The signer computes $V = (r + h)S_{id}$ and sends it to the user.
 - **User (Unblinding):** The user computes $V' = \alpha V$. (U', V') is the blind signature on the message m .
- **Verify:** When receiving the blind signature (U', V') on the message m , the verifier accepts it if

$$\hat{e}_t(V', P_{pub} + H_1(id)P_1) = \hat{e}_t(U' + H_3(m, U')P_1, P_1) \quad (5.3)$$

5.3.3 Security Analysis

In this section, we show how the IBBS satisfies the blindness and unforgeability properties. Note the description in this section only discusses how the proposed

IBBS scheme satisfies some security requirements and is not a formal security proof.

Lemma 5.1. *The proposed IBBS scheme satisfies the requirement of the blindness property.*

Proof. Suppose $\mathcal{A}$ is a probabilistic polynomial-time algorithm which can control **Signer**. He has ability of extracting a secret key S_{id} from a user identity id . U_0 and U_1 are two honest users. m_0 and m_1 are two messages produced by $\mathcal{A}$. b is a random bit which is kept secret from $\mathcal{A}$. Now place m_b and m_{1-b} to the read-only input tape of U_0 and U_1 , respectively. $\mathcal{A}$ joins the blind signature issuing protocol with U_0 and U_1 in arbitrary order. If U_0 and U_1 output $\sigma(m_b)$ and $\sigma(m_{1-b})$ on their private tapes, where $\sigma(m_b)$ denotes the blind signature on the message m_b , then offer those outputs to $\mathcal{A}$. Otherwise, give $\perp$ to $\mathcal{A}$.

When $\mathcal{A}$ obtains $\perp$, the probability that $\mathcal{A}$ wins the game is equivalent to guessing the value of b randomly, i.e., the probability is $1/2$.

Suppose $\mathcal{A}$ obtains $\sigma(m_b)$ and $\sigma(m_{1-b})$ (where $\sigma(m_b)$ and $\sigma(m_{1-b})$ are (U'_j, V'_j) for $j \in \{0, 1\}$), instead of $\perp$. Let (U_i, h_i, V_i) (where $i \in \{0, 1\}$) denote the view during the blind signature issuing protocol. To prove blindness, it is sufficient to show that there always exists a unique pair of blinding factors α and β which map (U_i, h_i, V_i) to (U'_j, V'_j) for each $i, j \in \{0, 1\}$. We define $\alpha = \log_{V_i} V'_j$ (where ECDLP is assumed to be solved) and $\beta = h_i - (\log_{V_i} V'_j)^{-1} H_3(m, U'_j) \pmod{p}$. It is clear that both α and β are unique in $\mathbb{Z}_p^*$. Next, we show how two such unique blinding factors satisfy the following equation:

$$U'_j = \alpha U_i + \alpha \beta P_1 \tag{5.4}$$

From the non-degenerate property of the bilinear pairings, Equation 5.4 implies the correctness of the following equation:

$$\hat{e}_t(U'_j, P_1) = \hat{e}_t(\alpha U_i + \alpha \beta P_1, P_1) \quad (5.5)$$

We compute

$$\begin{aligned} & \hat{e}_t(\alpha U_i + \alpha \beta P_1, P_1) \\ = & \hat{e}_t(\log_{V_i} V'_j r_i P_1 + \log_{V_i} V'_j (h_i - (\log_{V_i} V'_j)^{-1} H_3(m, U'_j)) P_1, P_1) \\ = & \hat{e}_t(\log_{V_i} V'_j r_i P_1 + \log_{V_i} V'_j h_i P_1 - H_3(m, U'_j) P_1, P_1) \\ = & \hat{e}_t(\log_{V_i} V'_j (r_i + h_i) P_1, P_1) \hat{e}_t(H_3(m, U'_j) P_1, P_1)^{-1} \\ = & \hat{e}_t(\log_{V_i} V'_j (r_i + h_i) S_{id}, P_{pub} + H_1(id) P_1) \hat{e}_t(U_j + H_3(m, U'_j) P_1, P_1)^{-1} \hat{e}_t(U_j, P_1) \\ = & \hat{e}_t(\log_{V_i} V'_j V_i, P_{pub} + H_1(id) P_1) \hat{e}_t(U'_j + H_3(m, U'_j) P_1, P_1)^{-1} \hat{e}_t(U'_j, P_1) \\ = & \hat{e}_t(V'_j, P_{pub} + H_1(id) P_1) \hat{e}_t(U'_j + H_3(m, U'_j) P_1, P_1)^{-1} \hat{e}_t(U'_j, P_1) \end{aligned}$$

Because the pair (U'_j, V'_j) is a valid blind signature, then

$$\hat{e}_t(V'_j, P_{pub} + H_1(id) P_1) = \hat{e}_t(U'_j + H_3(m, U'_j) P_1, P_1)$$

Thus,

$$\hat{e}_t(\alpha U_i + \alpha \beta P_1, P_1) = \hat{e}_t(U'_j, P_1) \Leftrightarrow U'_j = \alpha U_i + \alpha \beta P_1$$

Regardless of the value of (U'_j, V'_j) and (U_i, h_i, V_i) , the existence of the blinding factors α and β always results in (U'_j, V'_j) and (U_i, h_i, V_i) having exactly the same relation defined by the blind signature issuing protocol. Therefore, the

probability that $\mathcal{A}$ wins the game is exactly $1/2$. $\square$

In the next paragraph, we prove the unforgeability of the proposed IBBS scheme.

Let $\mathcal{A}$ be an adversary who is a polynomial time algorithm controlling **User**. Given the public parameters $\{p, G_1, G_3, \hat{e}_t, H_1, H_3, P_1, P_{pub}\}$ and the user identity id , $\mathcal{A}$ tries to forge a valid message-signature pair without knowing the secret key of the user.

First, $\mathcal{A}$ performs identity attacks by querying **Extract**. Given the identity $\{id_1, id_2, \dots, id_{q_E}\}$ where $id_i \neq id$. **Extract** returns the corresponding secret keys $\{S_{id_1}, S_{id_2}, \dots, S_{id_{q_E}}\}$ to $\mathcal{A}$. If there exists a id_j such that $H_1(id_i) = H_1(id_j)$, then $\mathcal{A}$ has an ability of forging any valid signature. However, since $H_1(\cdot)$ is a random oracle, **Extract** only obtains a random number with a uniform distribution. $\mathcal{A}$ cannot obtain any useful information from the results of the query.

Suppose that $\mathcal{A}$ has interacted with the signer id . Let (h, U, V) denote the view in the blind signature issuing protocol. Because $V = (r + h)S_{id}$ is given and h is known, $\mathcal{A}$ must know the value of r if $\mathcal{A}$ wants to obtain the user's secret key S_{id} . But $\mathcal{A}$ only knows $U = rP_1$. Both U and P_1 are the elements of G_1 , computing r is equivalent to solve an ECDLP which is impossible. Thus $\mathcal{A}$ cannot obtain the user's secret key during the blind signature issuing protocol.

In the proposed IBS scheme in the random oracle model described in Chapter 4, $r = \omega^s = \hat{e}_t(sP_1, P_1)$ and $\omega^u = \hat{e}_t(H_2(m, r)P_1, P_1)$, the verifier accepts the signature if $\hat{e}_t(sP_1 + H_2(m, r)P_1, P_1) = \hat{e}_t(P_{pub} + id \cdot P_1, V)$. Compared with Equation 5.4, they are almost equivalent except two hash functions $H_2(\cdot)$ and

$H_3(\cdot)$. If $\mathcal{A}$ breaks the proposed IBBS scheme and forges (U', V') , then he must forge a valid signature of the proposed IBS scheme in the random oracle. From the proof of the security of the proposed IBS scheme in the random oracle model described in Chapter 4, $\mathcal{A}$ cannot forge the signature of the proposed IBBS scheme.

“One-more forgery” [149] is viewed as the most powerful attack on blind signature schemes. Similar to ZK-IBBS-II scheme, there is only one secret key in the new IBBS scheme, instead of multiple key components. The one-more forgery cannot be applied to the new IBBS scheme.

Remark 5.2. *The above security analysis is not a formal proof. In order to prove the security under the one-more forgery is secure, more work are still needed in future.*

5.3.4 Performance

In this section, we compare the performance of the proposed IBBS scheme and the two other IBBS schemes. In Table 5.3, the first column shows the operations that the proposed IBBS scheme needs. The second and third columns show the operations needed for the IBBS schemes, as proposed in [178] and in [179], respectively. I , M_{G_1} , $H_{M_{G_1}}$, Pa , A_{G_1} and M_{G_3} denote the cost of computing an inverse operation over $\mathbb{F}_p$, a scalar multiplication in G_1 , a *MapToPoint* function [36], a pairing, a point addition on G_1 and a multiplication on G_3 , respectively. From the timings of the cryptographic primitives shown in Table 4.1 of Chapter 4, the proposed IBBS scheme offers the fastest key extraction of all the three IBBS schemes. When the three blind signature issuing protocols are compared, the **User** of the proposed IBBS scheme needs $3M_{G_1} + 1A_{G_1}$, which is faster

Table 5.3: The comparison of the proposed IBBS scheme and other IBBS schemes

Schemes	The proposed IBBS scheme	ZK-IBBS-I scheme	ZK-IBBS-II scheme
Setup	$1M_{G_1}$	$1M_{G_1}$	$1M_{G_1}$
Extract	$1I + 1M_{G_1}$	$1H_M + 1M_{G_1}$	$1H_M + 1M_{G_1}$
User	$3M_{G_1} + 1A_{G_1}$	$3M_{G_1} + 1Pa$	$3M_{G_1} + 1A_{G_1} + 1H_M$
Signer	$2M_{G_1}$	$3M_{G_1}$	$2M_{G_1} + 1H_M$
Verify	$2Pa + 2M_{G_1}$	$2Pa + 1E + 1H_M$	$2Pa + 1M_{G_1} + 1H_M$
Public Parameters	$G_1, G_3, \hat{e}_t, H_1, H_3, P_1, P_{pub}$	$G_1, G_3, \hat{e}_t, H_2, H_3, H_{M_{G_1}}, P_1, P_{pub}$	$G_1, G_3, \hat{e}_t, H_1, H_3, H_{M_{G_1}}, P_1, P_{pub}$

Table 5.4: Timings of the proposed IBBS scheme and other IBBS schemes on $E_{ss2}^{1024}(\mathbb{F}_{512})$

Schemes	The proposed IBBS scheme	ZK-IBBS-I scheme		ZK-IBBS-II scheme	
	Time (ms)	Time (ms)	Impro.	Time (ms)	Impro.
Setup	5.01	5.01	0	5.01	0
Extract	5.03	20.46	75.4%	20.46	75.4%
User	15.06	35.31	57.3%	30.51	50.6%
Signer	10.02	15.03	33.3%	25.47	60.1%
Verify	50.58	57.93	12.7%	61.02	17.1%

than ZK-IBBS-I scheme ($3M_{G_1} + 1Pa$) and ZK-IBBS-II scheme ($3M_{G_1} + 1A_{G_1} + 1H_{M_{G_1}}$). Similarly, the **Signer** of the proposed IBBS scheme needs $2M_{G_1}$, which is fewer than ZK-IBBS-I scheme ($3M_{G_1}$) and ZK-IBBS-II scheme ($2M_{G_1} + 1H_{M_{G_1}}$). We conclude that the proposed IBBS scheme provides the most efficient blind signature issuing protocol in the three IBBS schemes.

In the **Verify** phase, ZK-IBBS-I scheme requires $2Pa + 1E + 1H_{M_{G_1}}$. ZK-IBBS-II scheme requires $2Pa + 1M_{G_1} + 1H_{M_{G_1}}$. The proposed IBBS scheme requires $2Pa + 2M_{G_1}$. We consider the case that three IBBS schemes are

implemented over $E_{ss2}^{1024}(F_{512})$. From Table 4.1 shown in Chapter 4, the timing for computing a *MapToPoint* hash function is longer than that for a scalar multiplication. Thus the proposed IBBS scheme provides the most efficient algorithm for verifying signatures. The experimental results over $E_{ss2}^{1024}(\mathbb{F}_{512})$ shown in Table 5.4 also provides the proof to validate the above description. The efficiency of signing a message is improved 33.3% compared with ZK-IBBS-I scheme and 60.1% compared with ZK-IBBS-II scheme.

We now compare the public parameters of the three IBBS schemes. H_{MG_1} represents the *MapToPoint* hash function in Table 5.3. It is obvious that the proposed IBBS scheme requires one less *MapToPoint* function than ZK-IBBS-I scheme and ZK-IBBS-II scheme. In addition, in the three blind signature issuing protocols, ZK-IBBS-I scheme requires the storage of at least six points and a pairing value: $P_1, Q_{id}, U, P_{pub}, S_{id}, V$ and t (as shown in Figure 5.1). ZK-IBBS-II scheme requires the storage of at least five points: P_1, Q_{id}, U, S_{id} and V (as shown in Figure 5.2). However, the proposed IBBS scheme only requires four points: P_1, U, S_{id} and V (as shown in Figure 5.3). Thus, the proposed IBBS scheme is more suitable in a limited resource environment compared to ZK-IBBS-I and ZK-IBBS-II schemes.

5.4 A New Identity-Based Chameleon Signature Scheme

5.4.1 Definition

Chameleon hash functions and chameleon signatures were proposed by Krawczyk and Rabin [110]. A chameleon hash function is a trapdoor one-way hash function which is collision-resistant for the signer but is collision-tractable for the recipient of the signature. Based on this non-standard collision-resistant hash function, chameleon signatures are constructed under a hash-then-sign paradigm and provide the property of non-transferability, i.e., no third party can accept a signature that has been issued to a designated recipient. In other words, chameleon signatures do not allow the recipient of a signature to disclose the contents of the signed information to any other third party without the signer's consent. Although undeniable signatures [59] also provide this property, they are either interactive or inefficient. Chameleon signatures are seen as a more feasible alternative to undeniable signatures.

Ateniese and Medeiros [9] proposed the first identity-based chameleon hash function (IBCHF) based on RSA and built the first IBCS scheme based on it. IBCS scheme is a combination of an IBCHF and an IBS scheme. It not only provides non-transferability, but also simplifies key management by using the user identity as the public key. Similar to the chameleon hash function, IBCHF is also a trapdoor one-way hash function. Ateniese and Medeiros [9] defined the identity-based chameleon hash function as follows:

Definition 5.5. *An identity-based chameleon hash function is composed of four*

algorithms: Setup, Extract, Hash and Forge.

- **Setup:** *This is a probabilistic algorithm. **TA** runs it to generate a master key MK and related public parameters PK .*
- **Extract:** *This is a probabilistic algorithm. Given the master key MK and the user identity id , it outputs the user's secret key S_{id} . In a chameleon hash function, S_{id} is also called the trapdoor information.*
- **Hash:** *This is a probabilistic algorithm. Given the related public parameters PK , the user identity id and a message m , it outputs a hash value h . In practice, the input of **Hash** includes a random element R . We use the following formula to denote this function:*

$$h = \text{Hash}(id, m, R)$$

- **Forge:** *This is a deterministic algorithm. Given the public parameters PK , the user identity id , the trapdoor information S_{id} , a message m' and a hash value h of a message m , it outputs a sequence of random bits that correspond to a valid computation of $\text{Hash}(id, m, R)$ yielding the target value h . It is denoted by $\text{Forge}(id, S_{id}, m, R, h, m')$*

IBCS scheme is defined as an IBS scheme computed over IBCHF of a message m under the intended recipient's identity [181].

Zhang et al. [181] introduced two identity-based chameleon hash functions from bilinear pairings: ZSS-IBCHF-I and ZSS-IBCHF-II¹. Both IBCHFs can

¹In Zhang et al.'s original paper [181], ZSS-IBCHF-I and ZSS-IBCHF-II are called Scheme 1 and Scheme 2, respectively.

cooperate with CC-IBS scheme [54] to construct IBCS schemes. Zhang et al. [181] only described IBCS scheme based on ZSS-IBCHF-I, referred to as ZSS-IBCS-I scheme in short. If ZSS-IBCS-II in ZSS-IBCS-I scheme is replaced by ZK-IBCHF-II, another IBCS scheme can be constructed. We call it ZSS-IBCS-II scheme.

5.4.2 Review of ZSS-IBCHF-II

In ZSS-IBCS-II scheme, **TA** uses two different key extraction algorithms to generate two secret keys for the same identity. The reason is that extracting the secret key associated with the identity of ZSS-IBCHF-II is different from doing it in the corresponding CC-IBS scheme. ZSS-IBCHF-II scheme requires the extraction of the secret key by $S_{id} = \frac{1}{s+H_1(id)}P_1$, where $H_1(\cdot)$ is a collision-resistant cryptographic hash function (e.g. SHA-2 hash function [136]). However, the corresponding IBS scheme extracts the secret key by $S_{id} = sH_M(id)$, where $s \in \mathbb{Z}_p^*$ is the master key, and $H_M(\cdot)$ is the so-called *MapToPoint* function. This is inefficient in terms of implementation.

Next, we review ZSS-IBCHF-II and correct an error of ZSS-IBCHF-II in [181]. Suppose $G_1 = E(\mathbb{F}_p)$ is a supersingular elliptic curve with the order q over the finite field $\mathbb{F}_p$ with $P_1 \in G_1$, let $\hat{e}_t(P_1, P_1)$ denote the modified Tate pairing. In the original ZSS-IBCHF-II, the chameleon hash algorithm is defined as

$$\text{Hash}(id, m, R) = \hat{e}_t(P_1, P_1)^{H_1(m)} \hat{e}_t(H_1(id) + P_{pub}, R)^{H_1(m)}$$

where $H_1(\cdot)$ is a general cryptographic hash function which maps a string $\{0, 1\}^*$

into an element of $\mathbb{Z}_p^*$. Thus, $H_1(id)$ is an element of $\mathbb{Z}_p^*$ while P_{pub} is an element of G_1 . It is impossible to compute the addition between an element of $\mathbb{Z}_p^*$ and an element of G_1 . In the following description of ZK-IBCHF-II, we have corrected this error.

- **Setup:** **TA** chooses a random element x from $\mathbb{Z}_p^*$ and computes $P_{pub} = xP$. $H_M : \{0, 1\}^* \rightarrow E(\mathbb{F}_p)$ is a *MapToPoint* function. $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$, is a collision-resistant hash function. **TA** publishes $\{p, G_1, G_3, \hat{e}_t, P_1, P_{pub}, H_1, H_M\}$ as the public parameters while x is kept as the master key.
- **Extract:** Given a user identity $id \in \mathbb{Z}_p^*$, **TA** computes the corresponding secret key $S_{id} = \frac{1}{x+H_1(id)}P_1$. **TA** will choose another x if $x + H_1(id) \equiv 0 \pmod{p}$.
- **Hash:** Given message $m \in \{0, 1\}^*$, the signer chooses a random element R from $E(\mathbb{F}_p)$, and computes the hash value by:

$$\text{Hash}(id, m, R) = \hat{e}_t(P_1, P_1)^{H_1(m)} \hat{e}_t(H_1(id)P_1 + P_{pub}, R)^{H_1(m)} \quad (5.6)$$

- **Forge:** The Forge algorithm is

$$\text{Forge}(id, S_{id}, m, R, m') = R' = H_1(m')^{-1}((H_1(m) - H_1(m'))S_{id} + H_1(m)R)$$

The corrected chameleon hash function is shown in Equation 5.6. We

re-deduce the forgery of ZK-IBCHF-II as follows:

$$\begin{aligned}
& \text{Hash}(id, m', R') \\
&= \hat{e}_t(P_1, P_1)^{H_1(m')} \hat{e}_t(H_1(id) \cdot P_1 + P_{pub}, R')^{H_1(m')} \\
&= \hat{e}_t(P_1, P_1)^{H_1(m')} \hat{e}_t(H_1(id)P_1 + P_{pub}, H_1(m')^{-1}((H_1(m) - H_1(m'))S_{id} + H_1(m)R))^{H_1(m')} \\
&= \hat{e}_t(P_1, P_1)^{H_1(m')} \hat{e}_t(H_1(id)P_1 + P_{pub}, (H_1(m) - H_1(m'))S_{id} + H_1(m)R) \\
&= \hat{e}_t(P_1, P_1)^{H_1(m')} \hat{e}_t(H_1(id)P_1 + P_{pub}, (H_1(m) - H_1(m'))S_{id}) \hat{e}_t(H_1(id)P_1 + P_{pub}, H_1(m)R) \\
&= \hat{e}_t(P_1, P_1)^{H_1(m')} \hat{e}_t(P_1, (H_1(m) - H_1(m'))P_1) \hat{e}_t(H_1(id)P_1 + P_{pub}, H_1(m)R) \\
&= \hat{e}_t(P_1, P_1)^{H_1(m)} \hat{e}_t(H_1(id)P_1 + P_{pub}, R)^{H_1(m)}
\end{aligned}$$

The modified chameleon hash function does not affect the security of ZK-IBCHF-II. Thus, Claims 2 and 3 described in [181] are still correct.

5.4.3 The Proposed IBCS Scheme

To combine with ZK-IBCHF-II, the identity id in the proposed IBS scheme in the random oracle model is redefined as a binary string, instead of an element of $\mathbb{Z}_p^*$. It is then mapped into an element of $\mathbb{Z}_p^*$ by a collision-resistant hash function $H_1(\cdot)$ which is described in ZK-IBCHF-II. This redefinition does not affect the security of the proposed IBS scheme in the random oracle model. Based on the proposed IBS scheme and the corrected ZSS-IBCHF-II, we construct a new IBCS scheme which is described as follows:

- **Setup:** **TA** picks a random element x from $\mathbb{Z}_p^*$, computes $P_{pub} = xP$ and precomputes $\omega = \hat{e}_t(P, P)$. x is the master key. The public parameters are $(p, G_1, G_3, \hat{e}_t, P_1, P_{pub}, \omega, H_1, H_2)$, where $H_2: \{0, 1\}^* \times G_3^* \rightarrow \mathbb{Z}_p^*$ is a hash function.

- **Extract:** Alice is the signer with the identity $id_A \in \{0,1\}^*$ and her secret key is $S_{id_A} = \frac{1}{x+H_1(id_A)}P_1$, Bob is the recipient with the identity $id_B \in \{0,1\}^*$ and the corresponding secret key is $S_{id_B} = \frac{1}{x+H_1(id_B)}P_1$.
- **Sign:** Given message m , Alice picks a random element s from $\mathbb{Z}_p^*$ and a random element R from G_1 , computes $r = \omega^s$, and

$$h = \text{Hash}(id_B, m, R) = \omega^{H_1(m)} \hat{e}_t(H_1(id_B) \cdot P_1 + P_{pub}, R)^{H_1(m)}$$

Then, this is computed: $u = H_2(h, r)$ and $V = (u + s)S_{id_A}$. The signature (u, V, R) is sent to the verifier.

- **Verify:** The verifier computes $r = \hat{e}_t(P_{pub} + H_1(id_A) \cdot P_1, V)\omega^{-u}$, and accepts the signature if

$$u = H_2(\text{Hash}(id_B, m, R), r)$$

Where the function **Hash** represents ZK-IBCHF-II.

Bob can use a different message m' to forge an element $R' \in E(\mathbb{F}_p)$ such that $\text{Hash}(id_B, m', R') = \text{Hash}(id_B, m, R)$. Thus, Bob cannot prove to others that the message that Alice actually signed is not m' but m . So, the proposed IBCS scheme satisfies the non-transferability requirement. The unforgeability of ZK-IBCS-I scheme depends on the security of CC-IBS scheme and ZK-IBCHF-I. Similarly, the unforgeability of the new chameleon signature scheme also depends on the security of the proposed IBS scheme in the random oracle model and ZSS-IBCHF-II.

We notice that ZSS-IBCS-II scheme needs two different secret key extraction algorithms. One is $S_{id} = sQ_{id}$ (where Q_{id} is a point on the elliptic curve generated by the *MapToPoint* hash function) for CC-IBS scheme, while the other is $S_{id} = \frac{1}{s+H_1(id)}P_1$ for ZSS-IBCHF. However, in the proposed IBCS scheme, one key extraction algorithm is enough. Both the proposed IBS scheme in the random oracle model and ZK-IBCHF use $S_{id} = \frac{1}{s+H_1(id)}P_1$ to extract secret key.

5.4.4 Performance

Besides the advantage of having one less key extraction algorithm, the proposed IBCS scheme also provides more efficient **Sign** and **Verify** algorithms than ZK-IBCS-I and ZK-IBCS-II schemes. Table 5.5 lists the operations that the proposed IBCS scheme and ZK-IBCS schemes need in the **Sign** and **Verify** phases. We use the optimized method introduced in [181] to speed up the chameleon hash functions.

Let the random element $R = r'S_{id}$ where $r' \in_R \mathbb{Z}_p^*$, $\tau = \hat{e}_t(H_M(id), P_{pub})$. Then, ZK-IBCHF-I only needs to compute $\text{Hash}(id, m, R) = \tau^{r'+H_1(m)}$, and ZK-IBCHF-I only needs to compute $\text{Hash}(id, m, R) = \omega^{(r'+1)H_1(m)}$. Both IBCHFs require 1 exponentiation on G_3 . Let M_{G_1} , E_{G_3} , Pa denote the cost of computing 1 scalar multiplication on G_1 , 1 exponentiation on G_3 and 1 pairing, respectively. From Table 5.5, in the sign phase, the proposed IBCS scheme requires one more exponentiation and one less scalar multiplication than two other IBCS schemes. When all three IBCS schemes are implemented on $E_{ss2}^{1024}(\mathbb{F}_{512})$ or $E_{ns2}^{1024}(\mathbb{F}_{512})$, Table 4.1 described in Chapter 4 shows that the timing for computing a scalar multiplication on G_1 is longer than computing an exponentiation on G_3 . Thus, **Sign** of the proposed IBCS scheme is more efficient than ZK-IBCS schemes. In

Table 5.5: The comparison of the proposed IBCS scheme and ZK-IBCS schemes

Schemes	The proposed IBCS scheme	ZK-IBCS-I scheme	ZK-IBCS-II scheme
Sign	$1M_{G_1} + 2E_{G_3}$	$2M_{G_1} + 1E_{G_3}$	$2M_{G_1} + 1E_{G_3}$
Verify	$1Pa + 1M_{G_1} + 2E_{G_3}$	$2Pa + 1M_{G_1} + 1E_{G_3}$	$2Pa + 1M_{G_1} + 1E_{G_3}$

Table 5.6: Timings of the proposed IBCS scheme and ZK-IBCS schemes on $E_{ss2}^{1024}(\mathbb{F}_{512})$

Schemes	The proposed IBCS scheme	ZK-IBCS-I scheme		ZK-IBCS-II scheme	
	Time (<i>ms</i>)	Time (<i>ms</i>)	Impro.	Time (<i>ms</i>)	Impro.
Sign	8.85	11.94	25.9%	11.94	25.9%
Verify	29.13	49.41	41.0%	49.41	41.0%

the **Verify** phase, the proposed IBCS scheme requires one more exponentiation and one less pairing than ZK-IBCS schemes. Thus, even with any of the three IBCS scheme being implemented on any of the elliptic curves shown in Table 4.1, **Verify** of the proposed IBCS scheme is more efficient than ZK-IBCS schemes. The experimental results over $E_{ss2}^{1024}(\mathbb{F}_{512})$ shown in Table 5.6 also provides the proof to validate the above description. The improvement of the efficiency is significant. For example, compared with ZK-IBCS-I scheme, signing a message is improved 25.9% and verify a signature is improved 41%.

5.5 Summary

In Chapter 4, we introduced a new IBS scheme in the random oracle model. Building on that, this chapter developed some applications based on this new IBS scheme. The first application is Type 3 batch verification on the proposed IBS scheme, where batch verification is a method of verifying multiple signatures

simultaneously. Until now, only YCK-IBS scheme and the proposed IBS scheme support Type 3 batch verification. The experimental results show that Type 3 batch verification on the proposed IBS scheme in the random oracle model is much faster than that on YCK-IBS scheme. We then propose a new IBBS scheme over the proposed scheme and analyze its security. The performance analysis shows that the new IBBS scheme is a valid alternative to the existing IBBS schemes which is able to extract keys more quickly and provides better performance of the blind signature issuing protocol. Finally, we correct an error in an identity-based chameleon hash function of Zhang et al. [181], so that the proposed IBS scheme can cooperate with it to build a new chameleon signature scheme. The analysis shows that the proposed IBCS scheme is more efficient in the **Sign** and **Verify** phases than ZK-IBCS schemes.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This research work is motivated to address the problems of enhancing the performance of digital signature schemes such that they can be deployed in the environments with low resources such as wireless networks or mobile devices. The performance of the digital signature schemes are improved by implementing a new group signature scheme with shorter signature size and constructing some fast signature schemes. The proposed group signature scheme can shorten group signature size such that the transport delay can be reduced in wireless networks. The three new identity-based signature schemes efficiently reduce the public key cryptographic arithmetic operations in comparison with their counterparts. It not only benefits lowering power consuming in the low power consumer device, but also significantly shortens the time of signing messages and verifying signatures. The major results and contributions of this thesis are presented below.

In Chapter 3, a new practical group signature scheme in the random oracle model was proposed such that the delay caused by signature transport in wireless networks can be reduced. A trusted third party called the *security mediator* is introduced into group signatures. The appearance of the *security mediator* makes it efficient to implement a group signature scheme. Such a group signature scheme with a security mediator simplifies the procedure of joining groups, opening group signatures and revoking group membership immediately. The proposed group signature scheme is constructed based on the classical RSA problem. Similar to most efficient group schemes [8, 34], the proposed group signature scheme maintains the constancy of the group public key size. In addition, the length of the signature is also kept constant when new members join the group. We conclude by informally analyzing the security of the proposed group signature scheme, and show that the scheme meets all the requirements of a secure group signature scheme. More importantly, the proposed group signature scheme provides shorter signatures at 1024 bits security level than other group signature schemes. This will reduce the delay during the group signature transport in wireless networks.

In Chapter 4, a secure identity-based signature scheme using the random oracles is described such that the power consuming caused by the arithmetics of public key cryptography can be reduced. Its security depends on a variant of k -CAA. By eliminating the special hash function called the *MapToPoint* function, the proposed IBS scheme in the random oracle model improves the efficiency of extracting secret keys and verifying signatures. The experimental results in Table 4.3 - 4.6 indicate that, compared to all other existing bilinear pairings-based IBS schemes in the random oracle model, the proposed IBS scheme provides the most efficient key extraction and verification at the 1024 bits and 2048 bits

security levels. Verification is faster by at least 50% on a supersingular elliptic curve with embedding degree 2. The second aspect of Chapter 4 is that a simple method for building suitable non-supersingular elliptic curves is introduced in this chapter. The fast signing messages and verifying signatures efficiently reduce the power consuming in cell phones and PDAs.

In Chapter 5, three applications of the proposed IBS scheme in the random oracle model are presented for lower the power consuming in the mobile devices due to reducing the arithmetic of public key cryptography. We show how the proposed scheme supports batch verification, *i.e.* the verifier can verify multiple signatures simultaneously instead of verifying one signature at a time. The experimental results depict that applying Type 3 batch verification of the proposed IBS scheme in the random oracle model significantly reduces the time taken to verify the signatures. Compared with the use of Type 3 batch verification on YCK-IBS scheme, Type 3 batch verification of the proposed IBS scheme in the random oracle model has at least 80% savings when the number of signatures to be verified is more than 100. We build on the proposed IBS scheme in the random oracle model and introduce a new IBBS scheme. The simple security analysis carried out indicates that the proposed IBBS scheme is more efficient than other IBBS schemes. It extracts keys more quickly than other IBBS schemes, thus reducing the expense incurred by the trusted authority for key generation. Furthermore, it provides more efficient blind signature issuing protocol. By combining the proposed IBS scheme in the random oracle model and an existing IBCHF described by [181], we successfully construct a new IBCS scheme. The performance analysis shows that the new IBCS scheme is more efficient than other existing IBCS schemes.

6.2 Recommendations for Further Research

The results presented in this thesis suggest a series of interesting questions for future research. We present possible avenues for further investigation below.

6.2.1 Secure Short Group Signature

There remains a great deal of research potential in the proposed group signature scheme. Firstly, the given security analysis is informal and there are some overlapping requirements in the security analysis. We expect to propose a formal security notion, similar to the one mentioned by Bellare et al. [26], such that the proposed group signature scheme can be proven secure under this security notion. Secondly, the proposed group signature scheme constructed here is based on the classical RSA signature. With the aid of a trusted third party called a security mediator, we use two authorities to construct a group signature scheme. This new group signature scheme provides shorter signatures at the security level of 1024 bits than other group signature schemes. In future, we hope to explore the use of other regular signature schemes based on DLP or ECDLP, including signature schemes using bilinear pairings, for constructing group signature schemes with two authorities. To date, the shortest signature scheme is based on the bilinear pairings. Thus, we believe that we can construct a group signature scheme with a shorter signature than the scheme proposed in this thesis. Research on the use of two authorities in a group signature scheme is nascent, and there was only little discussion of the prior literature. In future, we hope not only to discuss the advantage of using two authorities in a group signature scheme, but also to consider potential attacks on two authorities.

6.2.2 Batch Verification on IBS Scheme Without Random Oracles

Chapter 4 introduced an IBS scheme that uses random oracles and performs well at the 1024/2048 bits security level. In Chapter 5, we investigate how Type 3 batch verification could be supported by the proposed IBS scheme. The experimental results show that batch verification significantly shortens the time for verifying multiple signatures. The greater the number of signatures to be verified, the more time taken is reduced. Thus, batch verification is viewed as a very useful method when multiple signatures need to be verified. However, no IBS scheme in the standard model cannot support all types of batch verification. To deploy the IBS scheme in environments which do not require hash functions, we plan to construct an efficient IBS scheme in the standard model such that it supports batch verification. To our best knowledge, no such IBS scheme has been proposed in the existing literature. This challenging topic is worthy of further investigation.

6.2.3 Secure IBBS Scheme and Partial IBBS Scheme

There remains a great deal of research potential in the studies of the IBBS scheme. “One-more forgery” is still the most effective attack method for IBBS scheme. All three IBBS schemes described in Chapter 5 lack provide a formal proof that they will be secure against one-more forgery attacks. Thus, there is a need to further analyze the security of the proposed IBBS scheme under one-more forgery attacks. We also suggest further investigation of the partial IBBS scheme from the bilinear pairings. By using the proposed IBS scheme in

the random oracle model, the proposed IBBS provides better performance than other IBBS schemes. How to extend the proposed IBS scheme in the random oracle model into a partial IBBS scheme is an interesting topic. Furthermore, we constructed an IBS scheme without using random oracles in Chapter 4. Motivated by it, an IBBS scheme without random oracles should be considered in future work. We are seeking suggestions on the problems we have mentioned.

6.2.4 More Identity-Based Signature Schemes

Building on the proposed IBS scheme in the random oracle model, we constructed a new IBBS scheme and a new IBCS scheme. These schemes are more efficient than the corresponding counterparts. Researchers have proposed many identity-based digital signature schemes by combining existing IBS schemes with other signature schemes. In the future, we plan to investigate whether the proposed IBS schemes can be applied to other identity-based signature schemes, such as identity-based threshold signatures, identity-based proxy signatures, identity-based ring signatures and identity-based group signatures.

It is hoped that the results obtained in this thesis will help to establish a basis for future work towards more secure and efficient implementations of digital signature schemes.

Author's Publications

Published and accepted papers

1. Shi Cui, Pu Duan and Choong Wah Chan, "A new method of building more non-supersingular curves." In: *Proceedings of 2005 ICES, International Workshop on Information Security and Hiding (ISH'05)*, Singapore. LNCS 3481, pp. 657-664, Springer-Verlag, Singapore, May 2005.
2. Shi Cui, Pu Duan and Choong Wah Chan, "A method of building more non-supersingular curves suitable for pairing-based cryptosystems." In: *IEICE Transaction On Fundamental of Electronics, Communications and Computer Sciences*, Vol.E88-A No. 9, pp. 2468-2470, 2005.
3. Shi Cui, Xiangguo Cheng and Choong Wah Chan, "Practical group signatures from RSA." *Proceedings of The IEEE 20th International Conference on Advanced Information Networking and Applications (AINA 2006)*, Vol. 1, pp. 211-215, Vienna, March 2006.
4. Shi Cui, Pu Duan, Choong Wah Chan, "An efficient identity-based signature scheme with batch verifications." *ACM International Conference Proceeding Series; Vol. 152, Proceedings of First International Conference on Scalable Information Systems (INFOSCALE 2006)*, ACM, HongKang, May 2006.
5. Shi Cui, Xiangguo Cheng, Choong Wah Chan and Pu Duan, "An efficient identity-based signature scheme and its applications", *International Journal of Network Security*, Vol. 5, No. 1. pp.89-98, July 2007.
6. Pu Duan, Shi Cui and Choong Wah Chan, "Finding more non-supersingular elliptic curves for pairing-based cryptosystems." In: *Proceedings of 2nd International Conference on Information Security (ICIS2005)*, Turkey, June 2005.

7. Pu Duan, Shi Cui and Choong Wah Chan, "Finding more non-supersingular elliptic curves for pairing-based cryptosystems." In: *International Journal of Information Technology*, Vol.2, pp. 157-163, 2005.
8. Pu Duan, Shi Cui and Choong Wah Chan, "Special polynomial families for generating more suitable elliptic curves for pairing-based cryptosystems." In: *Proceedings of 5th WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communications (EHAC 2006)*, Madrid, Feb 2006.
9. Pu Duan, Shi Cui and Choong Wah Chan, "New polynomial families for generating more suitable elliptic curve for pairing-based cryptosystems." In: *WSEAS Transactions on Computers*, vol.5, Issue 3, pp.528-535, March 2006.

Under review papers

1. Shi Cui, Pu Duan, Choong Wah Chan, "Identity-based signature with batch verification and identity-based blind signature." *Minor revised and re-submitted to the International Journal of Grid Computing: Theory, Methods and Applications*, Elsevier.
2. Shi Cui, Xiangguo Cheng, Choong Wah Chan "A short group signatures scheme from RSA." *Submitted to International Journal of Network Security*.

Bibliography

- [1] M. Abe and J. Camenisch, "Partially blind signature schemes." In: *Proceedings of SCIS*, 1997.
- [2] M. Abe and E. Fujisaki, "How to date blind signatures." In: *Proceedings of ASIACRYPT'96*, LNCS 1163, pp.244-252, Springer-Verlag, 1996.
- [3] M. Abe and T. Okamoto, "Provable secure partially blind signatures." In: *Proceedings of Crypt 2000*, LNCS 1880, pp.271-286, Springer-Verlag, 2000.
- [4] J. Arkko, T. Aura, J. Kempf, V. Mantyla, P. Nikander, and M. Roe, "Securing IPv6 neighbor discovery and router discovery." In: *Proceedings of WiSe 2002*, pp.77-86, ACM Press, 2002.
- [5] N. Asokan, V. Shoup and M. Waidner, "Optimistic fair exchange of digital signature." In: *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 4, pp 593-610, 2000.
- [6] G. Ateniese, D. Song and G. Tsudik, "Quasi-efficient revocation of group signatures." In: *Proceedings of FC 2002*, 2002.
- [7] G. Ateniese and G. Tsudik, "Some open issues and new directions in group signature schemes." In: *Proceedings of FC 1999*, LNCS 1648, pp.196-211, Springer-Verlag, 1999.
- [8] G. Ateniese and B. de Medeiros, "Efficient group signatures without trapdoors." In: *Proceedings of ASIACRYPT 2003*, LNCS 2894, pp.246-268, Springer-Verlag, 2003.
- [9] G. Ateniese and B. de Medeiros, "Identity-based chameleon hash and applications." In: *Proceedings of FC 2004*, 2004. Also in Cryptology ePrint Archive, Report 2003/167. Available from <http://eprint.iacr.org/2003/167>.

- [10] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme." In: *Proceedings of CRYPTO 2000*, LNCS 1880, pp.255-270, Springer-Verlag, 2000.
- [11] R. Balasubramanian and N. Koblitz, "The improbability that an elliptic curve hash sub-exponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm." In: *Journal of Cryptology*, Vol.11, pp.141-145, 1998.
- [12] Paulo S. L. M. Barreto, The pairing-based crypto lounge. Available from <http://paginas.terra.com.br/informatica/paulobarreto/pblounge.html>.
- [13] P. S. L. M. Barreto, H. Y. Kim, Ben Lynn and Michael Scott, "Efficient algorithms for pairing-Based Cryptosystems." In: *Proceedings of CRYPTO 2002*, LNCS 2442, pp.354-369, Springer-Verlag, 2002.
- [14] P. S. L. M. Barreto, B. Libert, N. McCullagh and J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps." In *Proceedings of ASIACRYPT 2005*, LNCS 3788, pp.515-532, Springer-Verlag, 2005.
- [15] P. S. L. M. Barreto, B. Lynn and M. Scott, "Constructing Elliptic Curves With Prescribed Embedding Degree." In: *Proceedings of SCN 2002*, LNCS 2576, pp.263-273, Springer-Verlag, 2002.
- [16] P. S. L. M. Barreto, B. Lynn and M. Scott, "Efficient algorithms for pairing-based cryptosystems." In: *Journal of Cryptology*, Vol. 17, No. 4, pp.321-334, Springer-Verlag, 2004.
- [17] P. S. L. M. Barreto, B. Lynn and M. Scott, "On the selection of pairing-friendly groups." In: *Proceedings of SAC 2003*, LNCS 3006, pp.17-25, Springer-Verlag, 2003.
- [18] P. S. L. M. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order." In: *Proceedings of SAC'2005*, LNCS 3897, pp.319-331, Springer-Verlag, 2006. See also Cryptology ePrint Archive, Report 2005/133. Available from <http://eprint.iacr.org/2005/133>.
- [19] M. Bellare, D. Micciancio and B. Warinschi, "Foundations of group signatures: formal definitions, simplified requirements, and a construction based on general assumptions." In: *Proceedings of EUROCRYPT 2003*, LNCS 2656, pp.614-629, Springer-Verlag, 2003.

- [20] M. Bellare, C. Namprempre and G. Neven, "Security proofs for identity-based identification and signature schemes." In: *Proceedings of EUROCRYPT 2004*, LNCS 3027, pp.268-286, Springer-Verlag, 2004.
- [21] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols." In: *Proceedings of the 1st CCCS*, pp.62-73, ACM Press, New York, 1993.
- [22] M. Bellare and P. Rogaway, "Optimal asymmetric encryption - how to encrypt with RSA." In: *Proceedings of EUROCRYPT'94*, LNCS 950, pp. 92-111, Springer-Verlag, 1994.
- [23] M. Bellare and P. Rogaway, "The exact security of digital signatures - how to sign with RSA and Rabin." In: *Proceedings of EUROCRYPT'96*, LNCS 1070, pp.399-416, Springer-Verlag, 1996.
- [24] M. Bellare, C. Namprempre, D. Pointcheval and M. Semanko, "The power of RSA inversion oracles and the security of Chaum's RSA-based blind signature scheme." In: *Proceedings of Financial Cryptography 2001*, LNCS 2339, pp.319-338, Springer-Verlag, 2001.
- [25] M. Bellare, J. Garay and T. Robin, "Fast batch verification for modular exponentiation and digital signatures." In: *Proceedings of EUROCRYPT'98*, LNCS 1403, pp.236-250, Springer-Verlag, 1998.
- [26] M. Bellare, H. Shi and C. Zhang, "Foundations of group signatures: the case of dynamic groups." In: *Proceedings of CT-RSA 2005*, LNCS 3376, pp.136-153, Springer-Verlag, 2005.
- [27] M. Beller and Y. Yacobi, "Batch Diffie-Hellman key agreement systems and their application to portable communications." In: *Proceedings of EUROCRYPT'92*, LNCS 658, pp.208-220, Springer-Verlag, 1992.
- [28] I. F. Blake, G. Seroussi and N. P. Smart, "Elliptic Curves in Cryptography." Cambridge University Press, 1999.
- [29] I. F. Blake, G. Seroussi and N. Smart, "Advances in Elliptic Curve Cryptography," London Mathematical Society Lecture Note Series. Cambridge University Press, 2005.
- [30] A. Boldyreva, "Efficient threshold signature, multisignature and blind signature schemes based on the Gap-Diffie-Hellman-Group signature scheme." In: *Proceedings of PKC 2003*, LNCS 2193, pp.31-46, Springer-Verlag, 2003.

- [31] D. Boneh and X. Boyen, "Efficient selective-ID secure identity based encryption without random oracles." In: *Proceedings of EUROCRYPT 2004*, LNCS 3027, pp.223-238, Springer-Verlag, 2004.
- [32] D. Boneh and X. Boyen, "Short signatures without random oracles." In: *Proceedings of EUROCRYPT 2004*, LNCS 3027, pp.56-73, Springer-Verlag, 2004.
- [33] D. Boneh and X. Boyen, "Secure Identity Based Encryption Without Random Oracles." In: *Proceedings of CRYPTO 2004*, LNCS 3152, pp.443-459, Springer-Verlag, 2004.
- [34] D. Boneh, X. Boyen and H. Shacham, "Short group signatures." In: *Proceedings of CRYPTO 2004*, LNCS 3152, pp.41-55, Springer-Verlag, 2004.
- [35] D. Boneh, X. Ding, G. Tsudik and C. M. Wong, "A method for fast revocation of public key certificate and security capabilities." In: *Proceedings of 10th USENIX Security Symposium*, pp. 297-308, 2001.
- [36] D. Boneh and M. Franklin, "Identity based encryption from the Weil pairing." In: *Proceedings of CRYPTO 2001*, LNCS 2139, pp.213-229, Springer-Verlag, 2001.
- [37] D. Boneh and M. Franklin, "Identity based encryption from the Weil pairing." In: *SIAM Journal of Computing*, Vol.32, No.3, pp.586-615, 2003.
- [38] D. Boneh, C. Gentry, B. Lynn and H. Shacham, "Aggregate and verifiably encrypted signature from bilinear maps." In: *Proceedings of EUROCRYPT 2003*, LNCS 2248, pp.514-532, Springer-Verlag, 2003.
- [39] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "A survey of two signature aggregation techniques." In: *Proceedings of CryptoBytes*, Vol. 6, No. 2, 2003.
- [40] D. Boneh, B. Lynn and H. Shacham, "Short signature from the Weil pairing." In: *Proceedings of ASIACRYPT 2001*, LNCS 2248, pp.514-532, Springer-Verlag, 2001.
- [41] D. Boneh, B. Lynn and H. Shacham, "Short signatures from the Weil pairing." In: *Journal of Cryptology*, Vol.17, No.4, pp.297-319, 2004.
- [42] J. Boyar, D. Chaum, I. B. Damgard and T.P. Pedersen, "Convertible undeniable signatures." In: *Proceedings of CRYPTO'90*, LNCS 537, pp.189-205, Springer-Verlag, 1991.

- [43] F. Brezing and A. Weng, "Elliptic curves suitable for pairing based cryptography." In: *Designs, Codes and Cryptography*, Vol. 37, No. 1, pp.133 - 141, 2005. See also Cryptology ePrint Archive, Report 2003/143. Available from <http://eprint.iacr.org/2003/143>.
- [44] S. Cui, P. Duan and C. Chan, "A new method of building more non-supersingular curves." In: *Proceedings of ISH'05*, LNCS 3481, pp.657-664, Springer-Verlag, 2005.
- [45] J. Camenisch, M. Koprowski and B. Warinschi, "Efficient blind signatures without random oracles." In: *Proceedings of SCN 2004*, LNCS 3352, pp.134-148, Springer-Verlag, 2005.
- [46] J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials." In: *Proceedings of CRYPTO 2002*, LNCS 2442, pp.61-76, Springer-Verlag, 2002.
- [47] J. Camenisch and M. Michels, "A group signature scheme with improved efficiency." In: *Proceedings of ASIACRYPT'98*, LNCS 1514, pp.160-174, Springer-Verlag, 1998.
- [48] J. Camenisch and M. Michels, "A group signature scheme based on an RSA-variant." *Technical Report RS-98-27*, BRICS, University of Aarhus, November 1998. An earlier version appears in [47].
- [49] J. Camenisch and M. Michels, "Separability and efficiency for generic group signature schemes." In: *Proceedings of CRYPTO'99*, LNCS 1666, pp.413-430, Springer-Verlag, 1999.
- [50] J. Camenisch and M. Stadler, "Efficient and generalized group signatures." In: *Proceedings of EUROCRYPT'97*, LNCS 1233, pp.465-479, Springer-Verlag, 1997.
- [51] J. Camenisch and M. Stadler, "Effient group signature schemes for large groups." In: *Proceedings of CRYPTO'97*, LNCS 1294, pp.410-424, Springer-Verlag, 1997.
- [52] R. Canetti, O. Goldreich and S. Halevi, "The random oracle methodology, revisited." In: *Proceedings of 30th Annual Symposium Theory of Computing*, ACM Press, 1998.
- [53] R. Canetti, O. Goldreich and S. Halevi, "The random oracle methodology, revisited." a new version of [52], 2002, available from http://www.math.weizmann.ac.il/~oded/p_rom.html.

- [54] J. C. Cha and J. H. Cheon, "An identity-based signature from gap Diffie-Hellman groups." In: *Proceedings of PKC 2003*, LNCS 2567, pp.18-30, Springer-Verlag, 2003.
- [55] T. K. Chan, K. Fung, J. K. Liu and V. K. Wei, "Blind spontaneous anonymous group signatures for ad hoc groups." In: *Proceedings of ESAS 2004*, LNCS 3313, pp.82-94, Springer-Verlag, 2004.
- [56] D. Chaum, "Blind signatures for untraceable payments." In: *Proceedings of CRYPTO'82*, pp.199-203, 1983.
- [57] D. Chaum, "Zero-knowledge undeniable signatures." In: *Proceedings of EUROCRYPT'90*, LNCS 473, pp.458-464, Springer-Verlag, 1990.
- [58] D. Chaum, "Designated confirmer signatures." In: *Proceedings of EUROCRYPT'94*, LNCS 950, pp.86C91, Springer-Verlag, 1995.
- [59] D. Chaum and H. van Antwerpen, "Undeniable signatures." In: *Proceedings of CRYPTO'89*, LNCS 435, pp.212-216, Springer-Verlag, 1989.
- [60] D. Chaum, E. van Heijst and B. Pfitzmann, "Cryptographically strong undeniable signatures, unconditionally secure for the signer." In: *Proceedings of CRYPTO'91*, LNCS 576, pp.470-484, Springer-Verlag, 1991.
- [61] D. Chaum and E. van Heyst, "Group Signatures." In: *Proceedings of EUROCRYPT'91*, LNCS 547, pp.257-265, Springer-Verlag, 1991.
- [62] L. Chen and C. Kudla, "Identity based authenticated key agreement from pairings." In: *Cryptology ePrint Archive*, Report 2002/184, 2002, , available from <http://eprint.iacr.org/2002/184>.
- [63] L. Chen and T. P. Pedersen, "New group signature schemes." In: *Proceedings of EUROCRYPT'94*, LNCS 950, pp.171-181, Springer-Verlag, 1994.
- [64] X. Chen, F. Zhang and K. Kim, "A new ID-based group signature scheme from bilinear pairings." In: *Proceedings of WISA 2003*, LNCS 2908, pp.585-592, Springer-Verlag, 2003.
- [65] X. Chen, F. Zhang and K. Kim, "ID-based Multi-proxy signature and blind multisignature from bilinear pairings." In: *Proceedings of KIISC*, pp.11-19, 2003.
- [66] H. Chien, J. Jan and Y. Tseng, "RSA based partially blind signature with low computation." In: *Proceedings of ICPADS 2001*, pp.385-389, 2001.

- [67] S. S. M. Chow, L. C. K. Hui, S. M. Yiu and K. P. Chow, "Forward-secure multisignature and blind signature schemes." In: *Applied Mathematics and Computation*, 2004.
- [68] S. S. M. Chow, L. C. K. Hui, S. Yiu and K. P. Chow, "Two improved partially blind signature schemes from bilinear pairings." In: *Proceedings of ACISP 2005*, LNCS 3574, pp.316-328, Springer-Verlag, 2005.
- [69] C. Cocks, "An identity based encryption scheme based on quadratic residues." In: *Proceedings of Cryptography and Coding*, LNCS 2260, Springer-Verlag, 2001.
- [70] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack." In: *Proceedings of Crypto'98*, LNCS 1462, pp.13-25, Springer-Verlag, 1998.
- [71] Y. Desmedt and Y. Frankel, "Shared generation of authentication and signature." In: *Proceedings of Crypto'91*, LNCS 576, , pp.457-469, Springer-Verlag, 1992.
- [72] W. Diffie and M. Hellman, "New directions in cryptography." In: *IEEE Transactions on Information Theory*, IT-22(6): pp.644-654, 1976.
- [73] S. Ding and G. Tsudik, "Simple identity-based cryptography with mediated RSA." In: *Proceedings of CT-RSA 2003*, LNCS 2612, pp. 193-210, Springer-Verlag, 2003.
- [74] X. Ding, G. Tsudik and S. Xu, "Leak-free group signatures with immediate revocation." In: *Proceedings of ICDCS 2004*, pp.608-615, IEEE Computer Society, 2004.
- [75] H. Dobbertin, A. Bosselaers and B. Preneel, "RIPEMD-160: a strengthened version of RIPEMD." In *Proceedings of the 3rd FSE*, LNCS 1039, pp.71-82, Springer-Verlag, 1996.
- [76] Y. Dodis, J. Katz, S. Xu and M. Yung, "Strong key-insulated signature schemes." In: *Proceedings of PKC 2003*, LNCS 2567, pp.130-144, Springer-Verlag, 2003.
- [77] D. N. Duc, J. H. Cheon and K. Kim, "A Forward-secure blind signature scheme based on the strong RSA assumption." In: *Proceedings of ICICS 2003*, LNCS 2836, pp.11-21, Springer-Verlag, 2003.

- [78] R. Dupont, A. Enge and F. Morain, "Building curves with arbitrary small MOV degree over finite prime fields." In: *Journal of Cryptology*, Vol. 18, No. 2, pp.79-89, Springer, 2005. See also Cryptology ePrint Archive, Report 2002/094. Available from <http://eprint.iacr.org/2002/094>.
- [79] R. Dutta, R. Barua and P. Sarkar, "Pairing-based cryptographic protocols : a survey." In: *Cryptology ePrint Archive*, Report 2004/064, 2004. Available from <http://eprint.iacr.org/2004/064>.
- [80] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms." In: *IEEE Transactions on Information Theory*, IT-31(4): pp.469-472, 1985.
- [81] S. Even, O. Goldreich and S. Micali, "On-line/Off-line digital signatures." In: *Proceedings of CRYPTO'89*, LNCS 435, pp.263-277, Springer-Verlag, 1990.
- [82] C. Fan and C. Lei, "Low-computation partially blind signature for electronic cash." In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E81-A(5), pp.818-824, 1998.
- [83] N. Ferguson, "Single term off-line coins." In: *Proceedings of EURO-CRYPT'93*, LNCS 765, pp.318-328, Springer-Verlag, 1993.
- [84] A. Fiat, "Batch RSA." In: *Journal of Cryptography*, Vol. 10, No. 2, pp.75-88, 1997.
- [85] A. Fiat and A. Shamir, "How to prove yourself: practical solutions to identification and signature problems." In: *Proceedings of CRYPTO'86*, LNCS 263, pp.186-194, Springer-Verlag, 1986.
- [86] G. Frey and H. G. Ruck, "A remark concerning m -divisibility and the discrete logarithm in the divisor class group of curves." In: *Mathematics of Computation*, Vol. 62, pp.865-874, 1994.
- [87] S. D. Galbraith, private communication.
- [88] S. D. Galbraith, K. Harrison and D. Soldera, "Implementing the Tate pairing" In: *Proceedings of ANTS V*, LNCS 2369, pp.324-337, Springer-Verlag, 2002.
- [89] S. D. Galbraith, J. McKee and P. Valenca, "Ordinary abelian varieties having small embedding degree." In: *Proceedings of a workshop on*

- Mathematical Problems and Techniques in Cryptology*, pp.29-45, 2005. See also Cryptology ePrint Archive, Report 2004/365. Available from <http://eprint.iacr.org/2004/365>.
- [90] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum and D. Boneh “Terra: A virtual machine-based platform for trusted computing.” In: *Proceedings of SOSP 2003*, pp.193-206, 2003.
 - [91] R. Gennaro, S. Halevi and T. Rabin, “Secure hash-and-sign signature without the random oracle.” In: *Proceedings of EUROCRYPT’99*, LNCS 1592, pp.123-139, Springer-Verlag, 1999.
 - [92] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, “Robust threshold DSS signatures.” In: *Proceedings of EUROCRYPT’96*, LNCS 1070, pp.354-371, Springer-Verlag, 1996.
 - [93] C. Gentry and A. Silverberg, “Hierarchical ID-based cryptography.” In: *Proceedings of ASIACRYPT 2002*, LNCS 2501, pp.548-566, Springer-Verlag, 2002.
 - [94] S. Goldwasser, S. Micali and C. Rackoff, “The knowledge complexity of interactive proof systems.” In: *SIAM journal on Computing*, Vol. 18, pp.186-208, 1989.
 - [95] S. Goldwasser, S. Micali and R. Rivest, “A digital signature scheme secure against adaptive chosen-message attacks.” In: *SIAM Journal of Computing*, 17(2), pp.281-308, 1988.
 - [96] R. Granger, D. Page and N. P. Smart, “High security pairing-based cryptography revisited.”, In: *Cryptology ePrint Archive*, Report 2006/059, 2006. Available from <http://eprint.iacr.org/2006/059>.
 - [97] L. C. Guillou and J. J. Quisquater, “A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory.” In: *Proceedings of EUROCRYPT’88*, LNCS 330, pp.123-128, Springer-Verlag, 1988.
 - [98] L. C. Guillou and J. J. Quisquater, “A “paradoxical” identity-based signature scheme resulting from zero-knowledge.” In: *Proceedings of CRYPTO’88*, LNCS 403, pp.216-231, Springer-Verlag, 1989.
 - [99] D. Hankerson, A. Menezes and S. Vanstone, “Guide to Elliptic Curve Cryptography.” Springer, 2004.

- [100] F. Hess, "Efficient identity based signature schemes based on pairings." In: *Proceedings of SAC 2002*, LNCS 2595, pp.310-324, Springer-Verlag, 2003.
- [101] E. van Heyst and T. P. Pedersen, "How to make efficient fail-stop signatures." In: *Proceedings of EUROCRYPT'92*, LNCS 658, pp.366-377, Springer-Verlag, 1992.
- [102] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their applications." In: *Proceedings of EUROCRYPT'96*, LNCS 1070, pp.143-154, Springer-Verlag, 1996.
- [103] A. Joux, "A one-round protocol for tripartite Diffie-Hellman." In: *Proceedings of ANTS IV*, LNCS 1838, pp.385-394, Springer-Verlag, 2000.
- [104] W. S. Juang and C. L. Lei, "Blind threshold signatures based on discrete logarithm." In: *Proceedings of the 2nd Asian Computing Science Conference*, LNCS 1179, pp.172-181, Springer-Verlag, 1996.
- [105] W. Juang and C. Lei, "A secure and practical electronic voting scheme for real world environments." In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E80-A(1), pp.64-71, 1997.
- [106] A. Juels, M. Luby and R. Ostrosky, "Security of blind digital signatures." In: *Proceedings of CRYPTO'97*, LNCS 1294, pp.150-164, Springer-Verlag, 1997.
- [107] J. Kim, K. Kim and C. Lee, "An efficient and provable secure threshold blind signature." In: *Proceedings of ISISC 2001*, LNCS 2288, pp.318-327, Springer-Verlag, 2002.
- [108] N. Koblitz, "Elliptic curve cryposystems." In: *Mathematics of Computation*, Vol.48, No.5, pp.203-209, 1987.
- [109] N. Koblitz and A. Menezes, "Pairing-based cryptography at high security levels." In: *Cryptography and Coding: 10th IMA International Conference*, LNCS 3796, pp.13-36, Springer-Verlag, 2005.
- [110] H. Krawczyk and T. Rabin, "Chameleon hashing and signatures." In: *Proceedings of NDSS 2000*, pp.143-154, 2000.

- [111] K. Kurosawa and S. Heng, "From digital signature to ID-based identification/signature." In: *Proceedings of PKC 2004*, LNCS 2947, pp.248-261, Springer-Verlag, 2004.
- [112] IEEE P1556 Working Group, VSC Project. Dedicated short range communications (DSRC), 2003.
- [113] IEEE 1363 Working Group, Standard Specifications For Public Key Cryptography - IEEE Std 1363-2000, 2000.
- [114] International Organization for Standardization, ISO 9796 Digital signature schemes giving message recovery, 2002.
- [115] International Telecommunication Union, Rec. X.509 (revised) the dictionary - authentication framework, 1993.
- [116] K. Itakura and K. Nakamura, "A public-key cryptosystem suitable for digital multisignatures." In: *NEC Research and Development*, Vol. 71, pp.1-8, 1983.
- [117] L. Lamport, "Constructing digital signatures from one-way function." In: *Technical Report SRI-CSL-98*, SRI International, 1979.
- [118] B. Libert and J. Quisquater, "The exact security of an identity based signature and its applications." In: *Cryptology ePrint Archive*, Report 2004/102, 2004. Available from <http://eprint.iacr.org/2004/102>.
- [119] B. Libert and J. Quisquater, "Identity based undeniable signatures." In: *Proceedings of CT-RSA 2004*, LNCS 2964, pp.112-125, Springer-Verlag, 2004.
- [120] H. Lipmaa, Cryptographic Signature Schemes. Available from <http://www.cs.ut.ee/~lipmaa/crypto/link/signature/>
- [121] L. Liu and Z. Cao, "Universal forgeability of a forward-secure blind signature scheme proposed by Duc et al." In: *Cryptology ePrint Archive*, Report 2004/262, 2004. Available from <http://eprint.iacr.org/2004/262>.
- [122] A. Lysyanskaya and Z. Ramzan, "Group blind digital signatures: a scalable solution to electronic cash." In: *Proceedings of FC 1998*, LNCS 1465, pp.184-197, Springer-Verlag, 1998.

- [123] M. Mambo, K. Usuda and E. Okamoto, "Proxy signatures: Delegation of the power to sign messages." In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E79-A(9), pp.1338-1353, 1996.
- [124] W. Mao, "Modern cryptography: theory and practice." Prentice Hall , 2003.
- [125] A. Menezes, T. Okamoto and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field." In: *IEEE transaction on Information Theory*, 39, pp.1639-1646, 1993.
- [126] S. Micali, K. Ohta and L. Reyzin, "Accountable-subgroup multisignatures." In: *Proceedings of CCS 2001*, pp.245-254, ACM Press, 2001.
- [127] S. Micali and R. Rivest, "Transitive signature schemes." In: *Proceedings of CT-RSA 2002*, LNCS 2271, pp.236-243, Springer-Verlag, 2002.
- [128] V. Miller, "Use of elliptic curves in cryptography." In *Proceedings of CRYPTO'85*, LNCS 218, pp.417-426, Springer-Verlag, 1986.
- [129] S. Mitsunari, R. Sakai and M. Kasahara, "A new traitor tracing." In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E85-A(2), pp.481-484, 2002.
- [130] A. Miyaji, M. Nakabayashi and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction." In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E84-A(5), pp.1234-1243, May 2001.
- [131] S. Miyazaki, M. Abe and K. Sakurai, "Partially blind signature schemes for the DSS and for a discrete log based message recovery signature." In: *Proceedings of Koera-Japan Joint Workshop on Information Security and Cryptology*, pp.217-226, 1997.
- [132] D. M'Raihi and D. Naccache, "Batch exponentiation - a fast DLP-based signature generation strategy." In: *Proceedings of 3rd ACM conference on Computer and Communications Security*, ACM Press, pp.58-61, 1996.
- [133] E. Mykletun, M. Narasimha and G. Tsudik, "Providing efficient data integrity mechanisms in outsourced databases." In: *Proceedings of NDSS 2004*, 2004.

- [134] D. Naccache, D. M'Raihi, S. Vaudenay and D. Raphaeli, "Can D.S.A. be improved? Complexity trade-offs with the digital signature standard." In: *Proceedings of EUROCRYPT'94*, LNCS 950, pp.77-85, Springer-Verlag, 1994.
- [135] National Institute of Standards and Technology, NIST FIPS PUB 186, Digital Signature Standard, U. S. Department of Commerce, May 1994.
- [136] National Institute of Standards and Technology, "FIPS 180-2: Secure Hash Standard (SHS)," 2002.
- [137] L. Nguyen and R. Safavi-Naini, "Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings." In: *Proceedings of ASIACRYPT 2004*, LNCS 3329, pp.372-386, Springer-Verlag, 2004.
- [138] A. M. Odlyzko, "Discrete logarithms: the past and the future." In: *Design, Code and Cryptography*, Vol.19, No.2, pp.129-145, 2002.
- [139] K. Ohta and T. Okamoto, "A digital multisignature scheme based on the Fiat-Shamir scheme." In: *Proceedings of ASIACRYPT'91*, LNCS 739, pp.75-79, Springer-Verlag, 1991.
- [140] K. Ohta and T. Okamoto, "Multi-signature schemes secure against active insider attacks." In: *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E82-A(1), pp.21-31, 1999.
- [141] T. Okamoto, "Provably secure and practical identification schemes and corresponding signature schemes." In: *Proceedings of CRYPTO'92*, LNCS 740, pp.31-53, Springer-Verlag, 1992.
- [142] K. G. Paterson, "ID-based signatures from pairings on elliptic curves." In: *Cryptology ePrint Archive*, Report 2002/003, 2002. Available from <http://eprint.iacr.org/2002/003>.
- [143] K. G. Paterson and G. Price, "A comparison between traditional PKIs and identity-based cryptography." In: *Information Security Technical Report* Vol.8, pp.57-72, 2003.
- [144] K. G. Paterson and J. C. N. Schuldt, "Efficient identity-based signature secure in the standard model." In: *Proceedings of ACISP 2006*, LNCS 4058, pp. 207-222, Springer-Verlag, 2006.

- [145] J. Pieprzyk, J. Seberry and Y. Zheng, "A one-way hashing algorithm with variable length and output ." In *Proceedings of AUSCRYPT'92*, LNCS 718, pp.83-104, Springer-Verlag, 1993.
- [146] D. Pointcheval and J. Stern, "Security proofs for signature schemes." In: *Proceedings of EUROCRYPT'96*, LNCS 1992, pp.387-398, Springer-Verlag, 1996.
- [147] D. Pointcheval and J. Stern, "Provable secure blind signature schemes." In: *Proceedings of ASIACRYPT'96*, LNCS 1163, pp.252-265, Springer-Verlag, 1996.
- [148] D. Pointcheval and J. Stern, "New blind signatures equivalent to factorization." In: *Proceedings of the 4th CCCS*, pp.92-99, ACM Press, 1997.
- [149] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures." In: *Journal of Cryptology*, Vol 13 No. 3, pp.361-396, Springer-Verlag, 2000.
- [150] M. Rabin, "Digital signatures and public-key functions as intractable as factorization." In: *MIT Lab of Computer Science, Technical Report*, MIT/LCS/TR-212, 1979.
- [151] R. L. Rivest, A. Shamir and L. M. Adleman. "A method for obtaining digital signatures and public-key cryptosystems." In: *Communications of the ACM*, 21(2) pp.120-126, 1978.
- [152] R. Rivest, A. Shamir and Y. Tauman, "How to leak a secret." In: *Proceedings of ASIACRYPT'01*, LNCS 2248, pp.552-565, Springer-Verlag, 2001.
- [153] R. Sakai, K. Ohgishi and M. Kasahara. "Cryptosystems based on pairing." In: *Proceedings of SCIS 2000*, 2000.
- [154] K. Sako, "Electronic voting scheme allowing open objection to the tally." In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E77-A(1), pp.24-30, 1994.
- [155] C. P. Schnorr, "Efficient identification and signatures for smart cards." In: *Proceedings of CRYPTO'89*, LNCS 435, pp.239-252, Springer-Verlag, 1990.
- [156] C. P. Schnorr, "Security of blind discrete log signatures against interactive attacks." In: *Proceedings of ICICS 2001*, LNCS 2229, pp.1-12, Springer-Verlag, 2001.

- [157] M. Scott, "Scaling security in pairing-based protocols." In: *Cryptography ePrint Archive*, Report 2005/139, 2005. Available from <http://eprint.iacr.org/2005/139>.
- [158] M. Scott and P. S. L. M. Barreto, "Generating more MNT elliptic curves." In: *Designs, Codes and Cryptography*, Vol. 38, No. 2, pp.209 - 217, 2006. See also Cryptology ePrint Archive, Report 2004/058, 2004. Available from <http://eprint.iacr.org/2004/058>. s
- [159] H. Shacham, "Sequential aggregate signatures from trapdoor homomorphic permutations." In: *Cryptography ePrint Archive*, Report 2003/091, 2003. Available from <http://eprint.iacr.org/2003/091>.
- [160] A. Shamir, "How to share a secret." In: *Communications of the ACM*, 22(11), pp.612-613, 1979.
- [161] A. Shamir, "Identity-based cryptosystems and signature schemes." In: *Proceedings of CRYPTO'84*, pp.47-53, Santa Barbara, CA, Aug. 1984.
- [162] Shamus Software Ltd. Miracl: Multiprecision integer and rational arithmetic C/C++ library, available from <http://indigo.ie/~mscott/>.
- [163] N. Smart, "An ID-based authenticated key agreement protocol based on the Weil pairing." In: *Electronic Letter*. Vol.38, No.13, pp.630-632, 2002.
- [164] Z. Tan, Z. Liu and C. Tang, "Digital proxy blind signature schemes based on DLP and ECDLP." In: *MM Research Preprints*, No. 21, pp.212-217, 2002.
- [165] Trusted Computing Group. Trusted Computing Platform Alliance (TCPA) Main Specification, 2003. Available from <http://www.trustedcomputinggroup.org>.
- [166] S. Vanstone, "Responses to NIST's proposal." In: *Communications of the ACM*, Vol.35, pp.50-52, 1992.
- [167] E. R. Verheul, "Evidence that XTR is more secure than supersingular elliptic curve cryptosystems." In *Proceedings of EUROCRYPT 2001*, LNCS 2045, pp.195-210, Springer-Verlag, 2001.
- [168] D. Liem Vo, F. Zhang and K. Kim, "A new threshold blind signature scheme from pairings." In: *Proceedings of SCIS 2003* Volume 1/2, pp.233-238, 2003.

- [169] G. Wang, Bibliography on Digital Signatures. Available from <http://www.i2r.a-star.edu.sg/icsd/staff/guilin/bible.htm>.
- [170] D. Wagner, "A generalized birthday problem." In: *Proceedings of CRYPTO 2002*, LNCS 2442, pp.288-303, Springer-Verlag, 2002.
- [171] L. C. Washington, "Elliptic curves: Number theory and cryptography." CRC Press, 2004.
- [172] B. Waters, "Efficient Identity-Based Encryption Without Random Oracles." In *Proceedings of EUROCRYPT 2005*, LNCS 3494, pp.114-127, Springer-Verlag, 2005.
- [173] S. Yen and C. Lai, "Improved Digital Signature suitable for batch verification." In: *IEEE Transaction on Computers*, Vol.44, No.7, pp.957-959, 1995.
- [174] X. Yi, "An identity-based signature scheme from the Weil pairing." In: *IEEE Communications Letters*, Vol.7, No.2, pp.76-78, 2003.
- [175] H. Yoon, J. H. Cheon and Y. Kim, "Batch verifications with ID-based signatures." In: *Proceedings of ICISC 2004*, LNCS 3506, pp.223-248, Springer-Verlag, 2005.
- [176] K. Zhang, "Threshold proxy signature schemes." In: *Proceedings of ISW'97*, LNCS 1396, pp.282-290, Springer-Verlag, 1997.
- [177] F. Zhang, X. Chen, W. Susilo and Y. Mu, "A new short signature scheme without random oracles from bilinear pairings." In: *Cryptology ePrint Archive*, Report 2005/386, 2005. Available from <http://eprint.iacr.org/2005/386>.
- [178] F. Zhang and K. Kim, "ID-based blind signature and ring signature from pairings." In: *Proceedings of ASIACRYPT 2002*, LNCS 2501, pp.533-547, Springer-Verlag, 2002.
- [179] F. Zhang and K. Kim, "ID-based blind signature and proxy signature from bilinear pairings." In: *Proceedings of ACISP 2003*, LNCS 2727, pp.312-323, Springer-Verlag, 2003.
- [180] F. Zhang, R. Safavi-Naini and C. Y. Lin, "New proxy signature, proxy blind signature and proxy ring signature schemes from bilinear pairings." In: *Cryptology ePrint Archive*, Report 2003/104, 2003, available from <http://eprint.iacr.org/2003/104>.

-
- [181] F. Zhang, R. Safavi-Naini and W. Susilo, “ID-based chameleon hashes from bilinear pairings.” In: *Cryptology ePrint Archive*, Report 2003/208, 2003. Available from <http://eprint.iacr.org/2003/208>.
 - [182] F. Zhang, R. Safavi-Naini and W. Susilo, “An efficient signature scheme from bilinear pairings and its applications.” In: *Proceedings of PKC 2004*, LNCS 2947, pp.277-290, Springer-Verlag, 2004.
 - [183] F. Zhang, R. Safavi-Naini and W. Susilo, “Efficient verifiably encrypted signature and partially blind signature from bilinear pairings.” In: *Proceedings of INDOCRYPT 2003*, LNCS 2904, pp.191-204, Springer-Verlag, 2003. Revised version is available from <http://eprint.iacr.org/2004/004>.
 - [184] Y. Zheng, “Digital signcryption or how to achieve cost (signature and encryption) $\ll$ cost (signature) + cost (encryption).” In: *Proceedings of CRYPTO’97*, LNCS 1294, pp.165-179, Springer-Verlag, 1997.

Appendix A

Group Law for Elliptic Curves

Let $E(K)$ be an elliptic curve over a finite field K . As we know, the set of points on $E(K)$ with a point at infinity $\mathcal{O}$ forms an abelian group under the group law of addition. Let $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be two different points on $E(K)$. The sum of P and Q is called point addition. It can be computed by a called chord-and-tangent rule which is described as follows: First draw a line l such that l passes through P and Q , and l will intersect the elliptic curve at a third point R' . Then the reflection R of R' about the x -axis is the sum of P and Q . If $P = Q$, then the sum of P and Q is called point doubling which is described as follows: First draw a line l such that l is tangent to the elliptic curve at P , and l will intersect the elliptic curve at a second point R' . Then the reflection R of R' about the x -axis is the double of P . Point addition and point doubling are depicted in Figure A.1 and A.2, respectively. if $P + Q = \mathcal{O}$, then Q is called the negative of P and is denoted by $-P$.

Group Law for Equation 2.2 on $\mathbb{F}_p$, where $p \neq 2, 3$.

Figure A.1: Point addition

Figure A.2: Point doubling

1. Negative. If $P = (x, y) \in E(\mathbb{F}_p)$, then the negative of P can be represented by $(x, -y)$.
2. Point addition. Let $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be two different points on $E(K)$, and $P \neq \pm Q$. Then the coordinate of the sum R of P and Q is (x_R, y_R) , where

$$x_R = \left(\frac{y_Q - y_P}{x_Q - x_P} \right)^2 - x_P - x_Q \quad \text{and} \quad y_R = \left(\frac{y_Q - y_P}{x_Q - x_P} \right)(x_P - x_R) - y_P$$

3. Point doubling. Let $P = (x_P, y_P)$ be a point on $E(K)$, and $P \neq \pm -P$. Then the point doubling $R = 2P = (x_R, y_R)$, where

$$x_R = \left(\frac{3x_P^2 + a}{2y_P} \right)^2 - 2x_P \quad \text{and} \quad y_R = \left(\frac{3x_P^2 + a}{2y_P} \right)(x_P - x_R) - y_P$$

Group Law for Equation 2.3 on $\mathbb{F}_{2^m}$

1. Negative. If $P = (x, y) \in E(\mathbb{F}_{2^m})$, then the negative of P can be represented by $(x, y + a_3)$.
2. Point addition. Let $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be two different points on $E(\mathbb{F}_{2^m})$, and $P \neq \pm Q$. Then the coordinate of the sum R of P and Q is (x_R, y_R) , where

$$x_R = \left(\frac{y_Q + y_P}{x_Q + x_P}\right)^2 + x_P + x_Q \quad \text{and} \quad y_R = \left(\frac{y_Q + y_P}{x_Q + x_P}\right)(x_P + x_R) + y_P + a_3$$

3. Point doubling. Let $P = (x_P, y_P)$ be a point on $E(\mathbb{F}_{2^m})$, and $P \neq \pm -P$. Then the point doubling $R = 2P = (x_R, y_R)$, where

$$x_R = \left(\frac{x_P^2 + a_4}{a_3}\right)^2 - 2x_P \quad \text{and} \quad y_R = \left(\frac{x_P^2 + a_4}{a_3}\right)(x_P + x_R) + y_P + a_3$$

Appendix B

Divisors and Miller's Algorithm

Let $E(K)$ be an elliptic curve defined over a finite field K . For each point $P \in E(K)$, (P) represents a formal symbol. A divisor D on $E(K)$ is a finite linear combination of such symbol with coefficients:

$$D = \sum_j a_j (P_j), \quad a_j \in \mathbb{Z}$$

The group of divisors is denoted by $\text{Div}(E)$ which is an additive group. The degree of a divisor is defined by

$$\deg(D) = \sum_j a_j \in \mathbb{Z}$$

The sum of divisors is defined by

$$\text{sum}(D) = \sum_j a_j P_j \in E(\bar{K})$$

All the divisors with degree 0 can form an important subgroup of $Div(E)$ which is denoted by $Div^0(E)$.

Suppose $E(K)$ is given by $y^2 = x^3 + Ax + B$, $f(x, y) \in \bar{K}(x, y)$ is a rational function on $E(K)$ which is defined as at least one point in $E(K)$. we say that f has a *zero* at $P \in E(K)$ if $f(P) = 0$, and f has a *pole* at $P \in E(K)$ if $f(P) = \infty$. The order of a rational function f at $P \in E(K)$ is defined to be the multiplicity of a *zero* at P and denoted by $ord_P(f)$. If f is a non-zero function on $E(K)$, the divisor of f is denoted by (f) which is defined as:

$$(f) = \sum_{P \in E(\bar{K})} ord_P(f)(P) \in Div(E)$$

If the degree of the divisor of a non-zero function f on $E(K)$ is always 0, such a divisor is called a *principal divisor*. If f has no zero or pole, then f must be a constant and $(f) = 0$. Suppose g is a non-zero function on $E(K)$. There is $(fg) = (f) + (g)$ and $(f/g) = (f) - (g)$.

Suppose D and D' are two divisors on $E(K)$. $D' \sim D$ means that D is equivalent to D' . If D and D' are equivalent, then there exists a function f such that $D' = D + (f)$. The group law on $Div(E)$ is similar to that on an elliptic curve and is described as follows: Suppose $l_1: E(K) \rightarrow K$ is a function on $E(K)$ which passes through two points P and Q on $E(K)$ (or is tangent to $E(K)$ on P) and hits $E(K)$ at some point $R'(x'_R, y'_R)$. The divisor (l_1) of the function l_1 can be represented by $(l_1) = (P) + (Q) + (R') - 3(\mathcal{O})$. Let $R = P + Q$. Then the vertical line $L_2(x) = (x - x'_R)$ must pass through the points R and R' as described in Appendix A. The divisor (l_2) of the function l_2 can be

represented by $(l_2) = (R) + (R') - 2(\mathcal{O})$. Therefore,

$$(l_1) - (l_2) = (P) + (Q) + (R') - 3(\mathcal{O}) - ((R) + (R') - 2(\mathcal{O}))$$

Reducing the redundant terms, the above equation can be deduced as:

$$(l_1/l_2) = (P) + (Q) - (R) - (\mathcal{O}) = (P) + (Q) - (P + Q) - (\mathcal{O}) \quad (\text{B.1})$$

The only method for computing the value of pairings is Miller Algorithm [29]. This algorithm uses the doubled-and-add method to construct a function f_q such that $(f_q) = q(P) - q(\mathcal{O})$. Suppose f_i is a non-zero-function on $E(K)$ such that its divisor (f_i) satisfies $(f_i) = i(P) - (iP) - (i - 1)(\mathcal{O})$. Miller Algorithm depends the following lemma [29]

Lemma B.1. *The function f_i can be chosen to satisfy the following conditions:*

1. $(f_1) = 1$.
2. Let l_1 and l_2 be the straight lines used in the computation of $iP + jP = (i + j)P$, then $f_{i+j} = f_i f_j \frac{l_1}{l_2}$.

Algorithm 2 shows the details of Miller Algorithm. Choosing a suitable point S is to compute $Q' = Q + S$ such that the divisor $(Q') - (S)$ is equivalent to $(Q) - (\mathcal{O})$. Thus, the support of $(Q') - (S)$ is disjoint to the support of (f) without any loss of generality.

Algorithm 2: Miller's Algorithm

Input: $P, Q \in E(K)$ where the order of P is q

Output: f_q

1. Choose a suitable point $S \in E(K)$;
 2. $Q' \leftarrow Q + S$;
 3. $T \leftarrow P$;
 4. $k \leftarrow \lfloor \log_2(q) \rfloor - 1$, $f \leftarrow 1$;
 5. While $k \geq 0$;
 6. Calculate lines l_1 and l_2 for doubling T ;
 7. $T \leftarrow (2)T$
 8. $f = f^2 \frac{l_1(Q')v(S)}{l_2(Q')l(S)}$;
 9. If the m th bit of q is one, then ;
 10. Calculate lines l_1 and l_2 for addition of T and P ;
 11. $T = T + P$;
 12. $f = f \frac{l_1(Q')v(S)}{l_2(Q')l(S)}$;
 13. $k \leftarrow k - 1$;
 14. Return f .
-

Appendix C

Elliptic Curves Parameters

The parameters of the supersingular elliptic curves $E_{ss2}^{1024}(\mathbb{F}_{512})$:

$$\begin{aligned} p &= 73132957625646785532203994141121553638406828962731283025431 \\ &\quad 02778210584118101444624864132462285921835023839111762785054 \\ &\quad 210425140241018649354445745491039387; \end{aligned}$$

$$a_4 = 1;$$

$$a_6 = 0;$$

$$\begin{aligned} h &= 10007920040268628970387373215664582404186858178692152430205 \\ &\quad 359413268619141100079249246263148037326528074908; \end{aligned}$$

$$q = 730750818665451459101842416358141509827966402561.$$

The parameters of the supersingular elliptic curves $E_{ns2}^{1024}(\mathbb{F}_{512})$:

$$\begin{aligned}
 p &= 73900125645670205661348905113630350506058561265270194279774 \\
 &\quad 52228620902865958117421136179239148255732551284306190056479 \\
 &\quad 417138297450573821810936941685437103 \\
 a_4 &= -3; \\
 a_6 &= 40466344126684204644990350029573925150019339944329164003328 \\
 &\quad 48257510823048494489340403061419732729989087630694029894581 \\
 &\quad 208721882447411493319862089156494766; \\
 h &= 10112903572332879862981885894607041268949114503781138821450 \\
 &\quad 155941567145701459710251995836379308501581937634 \\
 q &= 730750818665451459101842416358141509827966402561.
 \end{aligned}$$

The parameters of the supersingular elliptic curves $E_{ns6}^{1024}(\mathbb{F}_{157})$:

$$\begin{aligned}
 p &= 165707108867298088763888030961090962998151506177 \\
 a_4 &= -3; \\
 a_6 &= 152446285561183068325525053926770541662387929424; \\
 h &= 165707108867298088763888438032471518421242728401 \\
 q &= 165707108867298088763887623889710407575060283953.
 \end{aligned}$$