Identity Based Cryptography from Pairings

Yuen Tsz Hon

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

Identity Based Cryptography from Pairings

Yuen, Tsz Hon

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

Abstract

Identity based cryptography is a cryptosystem which uses identity, likes name, email address or other public information, as the public key. Digital certificate is not needed to prove the validity of the public key. This property is useful for promoting the use of cryptography in daily life, as the public key can be easily distributed and recognized by human.

Introduced by Shamir in 1984, identity based cryptography becomes a hot topic only since 2001, the introduction of identity based encryption using bilinear pairings over elliptic curves. Boneh and Franklin proposed that pairings are useful for constructing efficient identity based encryption scheme. Since then, there are lots of identity based cryptosystems proposed using pairings.

In this thesis, we introduce several schemes in identity based cryptography using pairings. Firstly, we introduce a new cryptosystem called blind identity based signcryption. We propose a new security model and an efficient instantiation for it. Secondly, we point out the deficiency of the existing definition of identity based group signatures in the literature. We then propose a precise definition and an efficient instantiation. Finally, we propose a hierarchical identity based signature without random oracles. As recently security proofs in the random oracle model receive certain doubts, we extend the study of identity based signatures with security proofs without using random oracle model.

概要

身份碼密碼系統的公鑰可以是使用者的身份碼，例如名字、電子郵件名稱或其它公眾能辨別的身份資訊，因此不需要電子證書來証明公鑰的合法性。由於身份碼密碼系統簡化公鑰的獲取和使用，所以它可以鼓勵更多人在互聯網上使用密碼系統。

身份碼密碼系統的概念是由 Shamir 在1984年首先提出。在2001年，Boneh 及 Franklin 提出一植基於橢圓曲線上的雙線性配對的身份碼密碼系統，是第一個較實際的身份碼加解密系統；之後，很多被提出的身份碼密碼系統，亦是植基於橢圓曲線上的雙線性配對。

本論文會提出三個植基於橢圓曲線上的雙線性配對的身份碼密碼系統。首先，本論文提出一個名為盲身份碼簽密的新密碼系統。本論文首先提出它的安全模型，然後建議一個實際的設計。此外，本論文指出已知的身份碼群體簽章的不足之處， 並提出一個更精確的定義，和一個相應的設計。最後，本論文提出一個不用隨機預言模型的分級身份碼簽章的設計。由於近年來利用隨機預言模型的安全證明備受質疑，所以本論文研究一些不需使用隨機預言模型的身份碼簽章。

Acknowledgement

First of all, I am profoundly grateful to Victor K. Wei, my advisor. He introduced the fascinating field of cryptography to me during my final year project of my undergraduate study. He helped me to push that paper into the CT-RSA conference, which inspired my ambition as an expert in cryptography. Besides his knowledge in the field, he also teaches me a lot in research methodology in these years. Many of the results reported herein are the fruits of lively discussions with him.

Next I wish to thank my colleagues. They are Allen Au, Patrick Chan, Rosanna Chan, Tony Chan, Sebastian Fleissner, Karyin Fung, Robert Leung, Joseph Liu and Patrick Tsang. We shared the joy and sadness in Information Security Lab every day. We enjoyed a game of Winning Eleven together, and we understood the difficulty faced by each other in the study. The time spent with my friends are the best time during my graduate study.

I am also thankful for Sherman Chow, Jin Li, Willy Susilo, Duncan Wong and Fangguo Zhang. It is a great experience to talk and work with them. I learned a lot by working with different people and it broadened my horizon.

Last, but certainly not least, I am grateful to my family for their continuous support. They give me a lot of supports during my study and encourage me to choose the way I want to go.

Contents

List of Notations

Below introduces the notations commonly used through out the rest of the thesis. Some notation will also be defined locally near its first use, while other notation will be used without further definition.

$S_1 \cup S_2$	union of sets S_1 and S_2
$S_1 \backslash S_2$	difference of sets S_1 and S_2
$S_1 \subseteq S_2$	S_1 is a subset of S_2
$x \in S, x \notin S$	element x (not in) set S
$x \in_R S$	sampling element x uniformly random in set S
$\mathbb{N}, \mathbb{Z}, \mathbb{R}$	sets of natural numbers, integers, and real numbers
$\mathbb{Z}^+$	set of positive integers
$\mathbb{Z}_n$	integers modulo n
$\mathbb{Z}_n^*$	multiplicative group of integers modulo n
$a \pmod b$	modulo operation: remainder of a divided by b
$\forall$	for all
$\exists$	there exists
$\mathrm{ord}(\mathbb{G})$	order of a group $\mathbb{G}$
$\Pr[E]$	probability of event E occurring
$E_1 \mid E_2$	event E_1 occurring given event E_2
$\lvert s \rvert$	number of elements in s if s is a finite set, or the length of s if s is a string, or the bit-length/size of s if s is an integer.
1^k	the string of k ones.
$s_1 \vert\vert s_2$	string s_1 concatenate with string s_2

Chapter 1

Introduction

"Who am I?" We often speak of one's "personal identity" as what makes one the person one is. Your identity in this sense consists roughly of those attributes that make you unique as an individual and different from others. Or it is the way you see or define yourself.

How do you identify someone? Usually we identify a person by his name, and also by what he did. However, different people may have different perception on the same person. For example, you may ask, "Who is Leonardo da Vinci?"

"He was a great painter." A seventh-grader said.

"He was the painter of the *Mona Lisa*." A visitor in Louvre Museum said.

"He was an Italian Renaissance artist who is good at painting and sculpture, with masterpieces including the *Mona Lisa*, the *Madonna of the Rocks* and the *Last Supper*." An university student with art major said.

"He was a master in anatomy, with thorough study of the human body in the *Vitruvian Man*." A medical student said.

"He was an inventor of a helicopter, a tank, the use of concentrated solar power, a calculator, a rudimentary theory of plate tectonics and many more." An engineer said.

“He was a representative of the Renaissance man: an architect, anatomist, sculptor, engineer, inventor, geometer, musician, and painter.” A historian said.

Therefore a person’s identity may differ from other’s point of view. Leonardo da Vinci may also be “the master of the Priory of Sion keeping the secret of the Holy Grail” according to the readers of the popular novel *The Da Vinci Code* [25].

In philosophy, personal identity is determined by the particular characteristics of the *self*. Self is an ancient subject in philosophy. Thales of Miletus (the first philosopher in the Greek tradition), when asked what was difficult, answered in a well-known apophthegm:

“To Know Thyself” ($\gamma\nu\varpi\theta\iota$ $\sigma\epsilon\alpha\mu\tau\sigma\nu$)

also attributed to Socrates, and inscribed on the Temple of Apollo at Delphi. In China, Lao Zi in his “Tao Te Ching” [62] says

Knowing others is wisdom.	(知人者智，)
Knowing the self is enlightenment.	(自知者明。)
Mastering others requires force.	(勝人者有力，)
Mastering the self requires strength.	(自勝者强。)

John Locke considered personal identity (or the self) to be founded on consciousness, and neither on the soul nor on the substance. This classic conception has been challenged by thinkers such as Marx, Nietzsche and Freud. The basis of personal identity is still the center of argument between philosophers. [94]

1.1 Identity Based Cryptography

In cryptography, it is usually divided into two categories: symmetric and assymetric cryptography. For asymmetric cryptography, there is a secret key which is only known by a party, and a public key which is known by the others. Therefore, asymmetric cryptography is also known as the public key cryptography. Usually, a public key is a string determined by the corresponding secret key. However, the distribution of public keys is a major problem in public key cryptography. It is troublesome and difficult to broadcast our public keys over the internet to all entities we want to communicate with. Moreover, even if we receive a public key, we are still not sure whether the public key is really associated with the entity we want to communicate with. Usually a Certificate Authority (CA) is needed to provide such an assurance, by providing digital certificates to the public keys. A high degree of trust is needed for the CA.

Knowing the user identity is important in cryptography. Identity based cryptography is a public key cryptosystem where the public key can be an arbitrary string, such as a name, an email address, or any combination of identities that can be used to identify a person uniquely. The identity can be chosen in a way that most communicating parties can recognize the key holder without ambiguity. For example:

pk = { Alice, CEO of ABCD Ltd., alice@abcd.com }

Then the distribution of public keys is no longer needed for identity based cryptography. It is different from the traditional public key cryptography that requires pre-computing key pairs and obtaining digital certificates for the public keys. This distinguishing characteristic of identity based cryptography is essential for some real world applications.

Identity based cryptosystem is a public key cryptosystem where the public key can be an arbitrary string such as an email address. It was firstly introduced by Shamir [83] in 1984. A trusted authority (TA) uses a master secret key to issue private keys to identities that request them.

For an Identity Based Encryption (IBE) scheme, Alice can securely encrypt a message to Bob using Bob's identity, such as bob@abcd.com, as the public key. There is no need for Alice to obtain Bob's public key certificate. Bob obtains his private key from the trusted authority and decrypts the ciphertext.

For an Identity Based Signature (IBS) scheme, Alice can sign a message using her private key that corresponds to an unambiguous identity of hers. Then anybody can verify the authenticity of the signature from her identity.

1.2 Hierarchical Identity Based Cryptosystem

Hierarchical identity based cryptosystem [50] is a generalization of identity based cryptosystem that mirrors the hierarchy of organizations. An identity at level ℓ of the hierarchy tree can issue private keys to its descendant identities at level $\ell + 1$, but cannot decrypt messages intended for other identities. In particular, an IBE and IBS is an 1-level hierarchical IBE and 1-level hierarchical IBS respectively.

Hierarchical identity based cryptosystem simulates the management system in a company. The CEO of a company monitors some senior managers. A senior manager monitors some managers and each manager monitors some employees. Hierarchical identity based cryptosystem emulates the hierarchy of organizations and the delegation of duties.

1.3 Our contributions

In this thesis, we introduce several schemes in identity based cryptography using pairings. We briefly introduce the main contributions of each scheme here. Details will be given in each corresponding chapter.

1. We introduce a new cryptosystem called blind identity based signcryption. We propose a new security model and an efficient instantiation for it.

2. We point out the deficiency of the existing definition of identity based group signatures in the literature. We then propose a precise definition, security model and an efficient instantiation.

3. We propose a hierarchical identity based signature without random oracles. The signature size is independent to the level of hierarchy. We also propose a constant-size hierarchical identity based signcryption without random oracles.

1.4 Publications

Three publications are produced directly from this thesis. The author has also produced other publications on cryptography during his MPhil study.

1.4.1 Publications Produced from This Thesis

1. Tsz Hon Yuen and Victor K. Wei. Fast and Proven Secure Blind Identity-Based Signcryption from Pairings. In *Proc. CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 305–322. Springer-Verlag, 2005. [96]

2. Victor K. Wei, Tsz Hon Yuen and Fangguo Zhang. Group Signatures where Group Manager, Members and Open Authority are Identity-Based. In *Proc. ACISP 2005*, volume 3574 of *Lecture Notes in Computer Science*, pages 468–480. Springer-Verlag, 2005. [93]

3. Tsz Hon Yuen and Victor K. Wei. Constant-Size Hierarchical Identity-Based Signature/Signcryption without Random Oracles. Cryptology ePrint Archive, Report 2005/412, 2005. [95]

1.4.2 Publications During Author's Study in the Degree

1. Sherman S.M. Chow, Tsz Hon Yuen, Lucas C.K. Hui and S.M. Yiu. Signcryption in Hierarchical Identity Based Cryptosystem. In *20th IFIP International Information Security Conference (SEC 2005)*, pages 443–457. Springer-Verlag, 2005. [40]

2. Sherman S.M. Chow, Joseph K. Liu, Victor K. Wei and Tsz Hon Yuen. Ring Signature without Random Oracles. In ACM Symposium on InformAtion, Computer and Communications Security(ASIACCS'06), Proceedings, pages 297-302, ACM Press, 2006. [38]

3. Victor K. Wei and Tsz Hon Yuen. More Short Signatures without Random Oracles. Cryptology ePrint Archive, Report 2005/463, 2005. [92]

1.5 Thesis Organization

In this chapter, we have discussed what is a personal identity, and its importance in various aspect. Then we introduce the use of identity in cryptography and the development of identity based cryptography. Finally we introduce the concept of hierarchical identity based cryptography, which is an extension of the concept of identity based cryptography.

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

Chapter 3 reviews several topics in identity based cryptography, that will be discussed in this thesis. We briefly introduce the basic idea and review the existing schemes in the literature.

Chapter 4 introduces a new cryptosystem called blind identity based signcryption. We propose a new security model for it. We then proceed to propose an identity based signcryption which is more efficient and secure than the existing schemes. Finally we extend the scheme to a new blind identity based signcryption scheme.

Chapter 5 points out the deficiency of the existing definition of identity based group signatures in the literature. We then propose a precise definition and a security model for it. Finally we give an efficient instantiation of identity based group signatures.

Chapter 6 proposes a constant-size hierarchical identity based signature without random oracles. We compare the advantages of our scheme with those of the existing schemes. Finally, we also propose a constant-size hierarchical identity based signcryption without random oracles.

In Chapter 7, we conclude the thesis.

Chapter 2

Background

Our goal in this chapter is to provide the necessary background and foundations of cryptography that will be used in the subsequent chapters. We first give an introduction to the topics of complexity theories, algebra, number theory. We then proceed to review some intractability assumptions that will be used in the subsequent chapters. Finally we review various cryptographic primitives including encryption, digital signatures, zero-knowledge proof of knowledge protocols, etc. Our main references are [52, 67, 85].

2.1 Complexity Theory

2.1.1 Order Notation

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

Definition 1 (Order Notation) $f(n) = O(g(n))$ *if there exists a positive constant* c *and a positive integer* n_0 *such that* $0 \leq f(n) \leq cg(n)$ *for all* $n \geq n_0$. $f(n) = \Omega(g(n))$ *if there exists a positive constant* c *and a positive integer* n_0 *such that* $0 \leq cg(n) \leq f(n)$ *for all* $n \geq n_0$. $f(n) = \Theta(g(n))$ *if there exists positive constant* c_1 *and* c_2*, and a positive integer* n_0 *such that* $c_1g(n) \leq f(n) \leq c_2g(n)$ *for all* $n \geq n_0$. $f(n) = o(g(n))$ *if for any positive* *constant $c > 0$ there exists a constant $n_0 > 0$ such that $0 \leq f(n) < cg(n)$ for all $n \geq n_0$.*

Definition 2 (Negligibility) *We call a function $\mu : \mathbb{N} \rightarrow \mathbb{R}$ negligible if for every positive polynomial $p(\cdot)$ there exists an N such that for all $n > N$, $\mu(n) < 1/p(n)$. A function is non-negligible if it is not negligible.*

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

2.1.2 Algorithms and Protocols

We model algorithms using Turing machines.

Definition 3 (Turing machine) *A Turing machine is a 7-tuple (Q, Σ, Γ, δ, q_0, q_{accept}, q_{reject}) where*

1. *Q is a finite set called* states,
2. *Σ is the input* alphabet *not containing the special blank symbol $\sqcup$,*
3. *Γ is the* tape alphabet, *where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$,*
4. *$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$ is the* transition function,
5. *$q_0 \in Q$ is the* start state, *and*
6. *$q_{accept} \subseteq Q$ is the* accept state.
7. *$q_{reject} \subseteq Q$ is the* reject state, *where $q_{accept} \neq q_{reject}$.*

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

Definition 4 (Algorithm) *A deterministic (resp. probabilistic) algorithm is a deterministic (resp. probabilistic) Turing machine.*

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

Given x, the output $A(x)$ of a probabilistic algorithm A is a random variable induced by the coin tosses. Let $A(x) = y$ denote the event "A outputs y on input x". By $Pr[A(x) = y]$, we mean the probability of this event. By $A(\cdot)$ we denote that the algorithm A has one input. By $A(\cdot, \ldots, \cdot)$ we denote that A has several inputs. $y \leftarrow A(x)$ denotes that y is obtained by running algorithm A on input x. In case A is deterministic, then this y is unique. If A is probabilistic (in which case we sometimes write $y \overset{R}{\leftarrow} A(x)$, then y is a random variable. If S is a set, then $y \leftarrow S$ (or sometimes $y \overset{R}{\leftarrow} S$) denotes that y was chosen form S uniformly at random.

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

Definition 5 (Efficient Algorithm) *An efficient algorithm or a polynomial time algorithm is an algorithm whose worst-case running time function is of the form $O(n^k)$, where n is the input size and k is a constant.*

We use the shorthand notation "PPT" for "probabilistic polynomial-time" when describing an algorithm. Next, we define what a two-party protocol is.

Definition 6 (Two-Party Protocol) *A two-party protocol is a pair of interactive probabilistic Turing machines $(\mathcal{P}, \mathcal{V})$. An execution (or run) of the* *protocol $(\mathcal{P}, \mathcal{V})$ on input x (for $\mathcal{P}$) and y (for $\mathcal{V}$) is an alternating sequence of $\mathcal{P}$-rounds and $\mathcal{V}$-rounds, each producing a message to be delivered to the other party (except for the last $\mathcal{V}$-round). The sequence of such message is called the transcript of this run of the protocol.*

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

2.1.3 Relations and Languages

If A is the set of all strings that machine M accepts, we say that A is the *language* of machine M and write L(M) = A.

A **verifier** for a language A is an algorithm V, where

$$A = \{w|V \text{ accepts } \langle w, c\rangle \text{ for some string } c\}.$$

Definition 7 (Polynomial Reducible) *We say that a language L is polynomially reducible to another language L_0 if there exists a deterministic polynomial-time-bounded Turing machine M which will convert each instance $I \in L$ into an instance $I_0 \in L_0$, such that $I \in L$ if and only if $I_0 \in L$.*

Definition 8 (Class P, NP, NPC) **P** *is the class of languages that are decidable in polynomial time on a deterministic single-tape Turing machine.* **NP** *is the class of languages that have polynomial time verifiers. A language B is* **NP***-complete if B is in* **NP***, and every A in* **NP** *is polynomially reducible to B. The class of all* **NP***-complete problems is denoted by* **NPC**.

2.2 Algebra and Number Theory

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

2.2.1 Groups

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

Definition 9 (Group) *A group $(G, \circ)$ is a set G together with an operation $\circ$ that satisfies:*

1. *Closure:* $\forall a, b \in G : a \circ b \in G$
2. *Associativity:* $\forall a, b, c \in G : a \circ (b \circ c) = (a \circ b) \circ c$
3. *Identity:* $\exists$ *unique element* $e \in G : \forall a \in G : a \circ e = e \circ a = a$.
4. *Inverse:* $\forall a \in G : \exists a^{-1} \in G : a \circ a^{-1} = a^{-1} \circ a = e$.

For simplicity, denote $a^i = \underbrace{a \circ a \circ \cdots \circ a}_{i}$. A group is *Abelian* if for all $a, b \in G$, $a \circ b = b \circ a$. For example, $(\mathbb{Z}, +)$ is an Abelian group.

Definition 10 (Group Order) *The order of the element $a \in G$ is the least positive integer i satisfying $a^i = e$, and its denoted by ord(a). If such an n does not exist, then the order of a is defined to be ∞.*

Definition 11 (Cyclic Group) *A group G is cyclic if there exists an element $a \in G$ such that for any $b \in G$, there exists an integer $i \geq 0$ such that $b = a^i$. Element a is called the generator (primitive root) of G. G is* *called the cyclic group generated by a. Order of the group G is the order of generator a.*

Definition 12 (Subgroup) *Let $(G, \circ)$ be a group. We say that $(H, \circ)$ is a subgroup of G if $H \subseteq G$ and $(H, \circ)$ is a group.*

Definition 13 *A ring $(R, +, \cdot)$ is a set R together with operations $+$ and $\cdot$ that satisfies:*

1. *$(R, +)$ is Abelian group, with 0 as identity.*
2. *$(R, \cdot)$ satisfies closure, associativity with identity 1, $1 \neq 0$.*
3. *Commutative: $\forall a, b \in G : a \cdot b = b \cdot a$.*
4. *Distribution: $\forall a, b, c \in G : a \cdot (b + c) = a \cdot b + a \cdot c$.*

Definition 14 *If the non-zero elements of a ring forms a group under multiplication, then the ring is called a field.*

Notice that $\mathbb{Z}_p$ is a finite field of prime order. We usually denote it as $\mathbb{F}_p$.

2.2.2 Elliptic Curve

Elliptic curves are defined over finite fields. For examples, in a prime field $\mathbb{F}_p$ with $p > 3$, we have a curve:

$$E : y^2 = x^3 + ax + b \pmod{p}$$

where a and b are constants satisfying $4a^3 + 27b^2 \neq 0 \pmod{p}$. Then we have points $P = (x, y)$ on the curve, together with $\mathcal{O} = (x, \infty)$ the point at infinity.

We denote E be an abelian group under the operation "+" defined as follows.

Definition 15 (Elliptic Curve Group Operation) *Let $P, Q \in E$, ℓ be the line containing P and Q (tangent line to E if $P = Q$), and R, the third point of intersection of ℓ with E. Let ℓ' be the line connecting R and $\mathcal{O}$. Then $P + Q$ is the point such that ℓ' intersects E at R, $\mathcal{O}$ and $P + Q$.*

We write $P + Q = -R$. Then we have the point multiplication for $k \in \mathbb{Z}$ defined as:

$$[k]P = \begin{cases} \underbrace{P + P + P}_{k} & \text{for } k > 0 \\ \mathcal{O} & \text{for } k = 0 \\ [-k](-P) & \text{for } k < 0 \end{cases}$$

2.2.3 Pairings

Original research by Menezes, Okamoto and Vanstone [71], and by Frey et al. [48], pointed out that the Weil and Tate pairings could be used for cryptanalytic purposes, undermining the security of certain types of elliptic curves. However this was followed by a prolonged hiatus before Sakai, Ohgishi and Kasahara [80] and Joux [59] independently observed that these very same condemned elliptic curves had in fact useful cryptographic properties. Soon Boneh and Franklin [18] came up with a very simple solution to the problem of identity based encryption using pairings.

Here we follow the notation in [20]. Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be (multiplicative) cyclic groups of prime order p. Let g_1 be a generator of $\mathbb{G}_1$ and g_2 be a generator of $\mathbb{G}_2$. We also let ψ be an efficiently computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, with $\psi(g_2) = g_1$. When $\mathbb{G}_1 = \mathbb{G}_2$ and $g_1 = g_2$ one could take ψ to be the identity map. On elliptic curves we can use the trace map as ψ. Then $\hat{e}$ is a bilinear map such that $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ with the following properties:

1. *Bilinearity*: For all $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}$, $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$.

	$E(\mathbb{F}_p)$ Tate pairing	$E(\mathbb{F}_p)$ Ate pairing
Pairing	2.9	3.1
Point Multiplication	3.0	1.1
Field Exponentiation	0.54	0.62
RSA Decryption	1.8	

Table 2.1: Timings in milliseconds on 3GHz Pentium IV, on 160-bit pairings and 1024-bit RSA.

2. *Non-degeneracy*: $\hat{e}(g_1, g_2) \neq 1$.

3. *Computability*: It is efficient to compute $\hat{e}(u, v)$ for all $u \in \mathbb{G}_1, v \in \mathbb{G}_2$.

It is believed that 160-bit pairings is as hard as 1024-bit RSA. Tate pairing can be computed faster than Weil pairing. Recently Ate pairing [57] is also proposed. [82] compares their computational efficiency as in table 2.1.

Sometimes we assume that $\mathbb{G}_1 = \mathbb{G}_2$ for simplicity. We will have a different definition for the pairing $\hat{e}$ in each chapter.

2.3 Intractability Assumptions

Various cryptographic protocols rely their security on the intractability of one or more mathematical problems in pairings $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$.

Definition 16 (co-CDH) *Let $g, g^a \in \mathbb{G}_1$ and $\bar{g} \in \mathbb{G}_2$ for an unknown integer $0 < a < p$. The co-Computational Diffie-Hellman Problem (co-CDH) is to find $\bar{g}^a$. An algorithm $\mathcal{A}$ has advantage ϵ in solving the co-CDH problem if* $\Pr[\mathcal{A}(g, g^a, \bar{g}) = \bar{g}^a] = \epsilon$. *We say that the (τ, ϵ)-co-CDH assumption holds if no τ-time algorithm has advantage at least ϵ in solving the co-CDH problem.*

Definition 17 (DDH) *Let g be a generator of a cyclic group $\mathbb{G}$ of prime order p. Let $g^a, g^b, T \in \mathbb{G}$ for some integers $0 < a, b < p$. The Decisional Diffie-Hellman Problem (DDH) is to distinguish T from g^{ab} and output 0/1. An algorithm $\mathcal{A}$ has advantage ϵ in solving the DDH problem if $|\Pr[\mathcal{A}(g, g^a, g^b, T) = 0] - \Pr[\mathcal{A}(g, g^a, g^b, g^{ab}) = 0]| = \epsilon$. We say that the (τ, ϵ)-DDH assumption holds in $\mathbb{G}$ if no τ-time algorithm has advantage at least ϵ in solving the DDH problem.*

By the bilinear property of pairings, the DDH problem in $\mathbb{G}_2$ is easy, while the co-CDH problem is believed to be hard.

Definition 18 (co-BDH) *Let $g, g^a, g^b \in \mathbb{G}_1$ and $\bar{g} \in \mathbb{G}_2$ for some unknown integers $0 < a, b < p$. The co-Bilinear Diffie-Hellman Problem (co-BDH) is to find $\hat{e}(g, \bar{g})^{ab}$. An algorithm $\mathcal{A}$ has advantage ϵ in solving the co-BDH problem if $\Pr[\mathcal{A}(g, g^a, g^b, \bar{g}) = \hat{e}(g, \bar{g})^{ab}] = \epsilon$. We say that the (τ, ϵ)-co-BDH assumption holds if no τ-time algorithm has advantage at least ϵ in solving the co-BDH problem.*

Definition 19 (co-DBDH) *Let $g, g^a, g^b \in \mathbb{G}_1$, $\bar{g} \in \mathbb{G}_2$, $R \in \mathbb{G}_T$ for some unknown integers $0 < a, b < p$. The co-Decisional Bilinear Diffie-Hellman Problem (co-DBDH) is to decide if $R = \hat{e}(g, \bar{g})^{ab}$ and output 0/1. An algorithm $\mathcal{A}$ has advantage ϵ in solving the co-DBDH problem if $|\Pr[\mathcal{A}(g, g^a, g^b, \bar{g}, R) = 0] - \Pr[\mathcal{A}(g, g^a, g^b, \bar{g}, \hat{e}(g, \bar{g})^{ab}) = 0]| = \epsilon$. We say that the (τ, ϵ)-co-DBDH assumption holds if no τ-time algorithm has advantage at least ϵ in solving the co-DBDH problem.*

Definition 20 (q-SDH) *The q-Strong Diffie-Hellman Problem (q-SDH) in $(\mathbb{G}_1, \mathbb{G}_2)$ is defined as follows: given a $(q+2)$-tuple $(g_1, g_2, g_2^x, \cdots, g_2^{x^q}) \in \mathbb{G}_1 \times \mathbb{G}_2^{q+1}$ for an unknown integer $0 < x < p$ with $g_1 = \psi(g_2)$ and g_2 is a random generator from $\mathbb{G}_2$, output a pair $(g_1^{1/(x+c)}, c)$ where $c \in \mathbb{Z}_p^*$. An algorithm $\mathcal{A}$*

has advantage ϵ in solving the q-SDH problem if $\Pr[\mathcal{A}(g_1, g_2, g_2^x, \cdots, g_2^{x^q}) = (g_1^{1/x+c}, c)] = \epsilon$. *We say that the (q, τ, ϵ)-SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$ if no τ-time algorithm has advantage at least ϵ in solving the q-SDH problem.*

The q-SDH problem is introduced in [14].

Definition 21 (k-SDH') *The k-Strong Diffie-Hellman' problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is as follows: given $g_1, {g_1}^{\gamma}, ..., {g_1}^{\gamma^k} \in \mathbb{G}_1$ and $g_2, {g_2}^{\gamma} \in \mathbb{G}_2$ as input, output a pair $({g_1}^{1/\gamma+x}, x)$ where $x \in \mathbb{Z}_p^*$.*

Definition 22 (k-CAA2) *The k-CAA2 problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is as follows: given $v, u \in \mathbb{G}_1$, $g_2, {g_2}^{\gamma} \in \mathbb{G}_2$ and pairs (A_i, e_i, λ_i) with distinct and nonzero e_i's satisfying $A_i^{\gamma+e_i} v^{\lambda_i} = u$ for $1 \leq i \leq k$ as input, output a pair $(A_{k+1}$, e_{k+1}, $\lambda_{k+1})$ satisfying $A_{k+1}^{\gamma+e_{k+1}} v^{\lambda_{k+1}} = u$, with $e_{k+1} \neq e_i$ for all $1 \leq i \leq k$.*

The above k-SDH' problem and k-CAA2 problem are proven equivalent in [91] assume the value $log_u(v)$ is known. [91] also shows that the k-SDH problem is at least as strong as the k-SDH' problem.

Definition 23 (ℓ-DHI) *The ℓ-Diffie-Hellman Inversion problem is that, given g, g^{α}, g^{α^2}, ..., $g^{\alpha^\ell} \in \mathbb{G}$, for unknown $\alpha \in \mathbb{Z}_p^*$, to compute $g^{1/\alpha}$.*

Definition 24 (ℓ-DHI*) *The ℓ-Diffie-Hellman Inversion * problem is that, given g, g^{α}, g^{α^2}, ..., $g^{\alpha^\ell} \in \mathbb{G}$, for unknown $\alpha \in \mathbb{Z}_p^*$, to compute $g^{\alpha^{\ell+1}}$.*

We say that the ℓ-DHI* assumption holds if no PPT algorithm can solve a random instance of the ℓ-DHI* problem with non-negligible probability. The ℓ-DHI problem and ℓ-DHI* problem are proven equivalent in [101].

We introduce the Decisional ℓ-wBDHI* problem in [15]. For simplicity, assume the pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.

Definition 25 (Decisional ℓ-wBDHI*) *The decisional ℓ-weak Bilinear Diffie-Hellman Inversion * problem is that, given g, h, g^{α}, $g^{\alpha^2}, \ldots, g^{\alpha^\ell} \in \mathbb{G}$ and $T \in \mathbb{G}_T$, for unknown $\alpha \in \mathbb{Z}_p^*$ and random generators g, h in $\mathbb{G}$, decide if $T = \hat{e}(g, h)^{\alpha^{\ell+1}}$.*

We say that the decisional ℓ-wBDHI* assumption holds if no PPT algorithm can solve a random instance of the decisional ℓ-wBDHI* problem with non-negligible probability over half.

2.4 Cryptographic Primitives

2.4.1 Public Key Encryption

A public key encryption scheme is a tuple $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ such that:

- $(ke, kd) \leftarrow \mathcal{G}(1^\lambda)$: On input a security parameter 1^λ, the key generation algorithm outputs an encryption key ke and a decryption key kd.
- $c \leftarrow \mathcal{E}(m, ke)$: On input a message m and an encryption key ke, the encryption algorithm $\mathcal{E}$ outputs a ciphertext c.
- $m \leftarrow \mathcal{D}(c, kd)$: On input a ciphertext c and a decryption key kd, the decryption algorithm $\mathcal{D}$ outputs a message m.

Correctness: We require that:

$$\mathcal{D}(\mathcal{E}(m, ke), kd) = m$$

The security goal of an encryption is defined as the indistinguishability of ciphertexts under adaptive chosen ciphertext attack (IND-CCA2). It is defined as the following game between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$:

1. (Setup.) $\mathcal{S}$ gives public key pk to $\mathcal{A}$.

2. (Probe 1.) $\mathcal{A}$ can make decryption queries to a decryption oracle $\mathcal{DO}$: On input a ciphertext c, output a message $m = \mathcal{D}(c, sk)$.

3. (Gauntlet.) $\mathcal{A}$ randomly picks two messages m_0, m_1 and gives to $\mathcal{S}$. $\mathcal{S}$ flips a coin b, computes $c^* = \mathcal{E}(m_b, pk)$ and returns c^* to $\mathcal{A}$.

4. (Probe 2.) $\mathcal{A}$ can makes decryption queries to a decryption oracle $\mathcal{DO}$, except for c^*.

5. (End Game.) $\mathcal{A}$ outputs a bit b'.

$\mathcal{A}$ wins the game if $b = b'$. $Adv_{\mathcal{A}}$ is $\mathcal{A}$'s probability of winning over half.

Definition 26 *An encryption scheme is indistinguishable against adaptive chosen ciphertext attack (IND-CCA2) if $Adv_{\mathcal{A}}$ is negligible.*

There exist some other weaker security models in which the attacker is given less power. If the above game has no Probe 2 phase, then the security becomes "indistinguishable against chosen ciphertext attack (IND-CCA1)". If the above game has no Probe 1 and 2 phase, then the security becomes "indistinguishable against chosen plaintext attack (IND-CPA)".

2.4.2 Digital Signature

A digital signature scheme is a tuple $(\mathcal{G}, \mathcal{S}, \mathcal{V})$ such that:

- $(sk, vk) \leftarrow \mathcal{G}(1^\lambda)$: On input a security parameter 1^λ, the key generation algorithm outputs a signing key ks and a verification key kv.

- $s \leftarrow \mathcal{S}(m, sk)$: On input a message m and a signing key sk, the signing algorithm $\mathcal{S}$ outputs a signature s.

- $\top/\bot \leftarrow \mathcal{V}(s, m, vk)$: On input a signature s, a message m and a verification key vk, the verification algorithm $\mathcal{V}$ outputs $\top$ for valid or $\bot$ for invalid.

Correctness: We require that:

$$\mathcal{V}(s, m, vk) = \begin{cases} \top \text{ with probability 1;} & \text{if } s \leftarrow \mathcal{S}(m, sk) \\ \perp \text{ with an overwhelming probability;} & \text{otherwise.} \end{cases}$$

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

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

The existential unforgeability against adaptive chosen plaintext attack of a signature scheme is defined as the following game between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$:

1. (Setup.) $\mathcal{S}$ gives public key pk to $\mathcal{A}$.

2. (Probe.) $\mathcal{A}$ can make signing queries to a signing oracle $\mathcal{SO}$: On input a message m, output a signature $s = \mathcal{S}(m, sk)$.

3. (End Game.) $\mathcal{A}$ outputs a signature s^* for message m^*.

$\mathcal{A}$ wins the game if $\mathcal{V}(s^*, m^*, pk) = \top$ and s^* is not the output from $\mathcal{SO}$. $Adv_{\mathcal{A}}$ is $\mathcal{A}$'s probability of winning.

Definition 27 *A signature scheme is existential unforgeable against adaptive chosen plaintext attack (EU-ACP) if $Adv_{\mathcal{A}}$ is negligible.*

2.4.3 Zero Knowledge

A zero-knowledge proof of knowledge or zero-knowledge (ZK) protocol is an interactive method for one party to prove to another that a (usually mathematical) statement is true, without revealing anything other than the veracity of the statement.

A zero-knowledge proof must satisfy three properties:

1. Completeness: if the statement is true, the honest verifier (that is, one following the protocol properly) will be convinced of this fact by an honest prover.

2. Soundness: if the statement is false, no cheating prover can convince the honest verifier that it is true, except with some small probability.

3. Zero-knowledge: if the statement is true, no cheating verifier learns anything other than this fact. This is formalized by showing that every cheating verifier has some simulator that, given only the statement to be proven (and no access to the prover), can produce a transcript that "looks like" an interaction between the honest prover and the cheating verifier.

The first two of these are properties of more general interactive proof systems. The third is what makes the proof zero-knowledge. In an interactive proof system, the sequence of information exchange is called a *proof transcript*.

ZK is subdivided into the following:

1. perfect ZK: if for any input, a proof transcript can be produced by a polynomial time algorithm with the same probability distribution.

2. honest-verifier ZK: if the verifier honestly follows the protocol, then the protocol is perfect ZK.

3. computational ZK: if for any input, a proof transcript can be simulated by a polynomial time algorithm with probability distribution which are polynomially indistinguishable from the real proof transcript.

4. statistical ZK: if for any input, a proof transcript can be simulated by a polynomial time algorithm with probability distribution which cannot be differentiated by any statistical distinguisher.

Σ-Protocol

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

Let $L \in \{0,1\}^*$ be a language. Let $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ be a polynomially bounded binary relation and let $L_{\mathcal{R}}$ be the language defined by $\mathcal{R}$. We require that a Σ-protocol satisfies:

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

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

2.5 Hash Functions

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

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

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

- *One-wayness.* For a given c, it is hard to find an x such that $H(x) = c$.
- *Weak collision resistance.* For a given x, it is hard to find an $x' \neq x$ such that $H(x) = H(x')$.
- *Strong collision resistance.* It is hard to find a pair $(x; x')$ with $x' \neq x$ such that $H(x) = H(x')$ if H is chosen at random from a family of hash-functions.

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

2.6 Random Oracle Model

The Random Oracle Model (ROM) is a paradigm that acts as a bridge between cryptographic theory and cryptographic practice. The idea of ROM was firstly formulated by Bellare and Rogaway [8].

Random oracle is a powerful and imaginary function that on any input, the distribution of the output hashed value is uniform in the function's output space. It has three properties: deterministic, efficient and uniform output. In the ROM, we assume hash functions are random functions and are publicly accessible by all parties. Random oracle, H, is an object to instantiate all hash functions in the model and reply all queries from the parties. A polynomial time algorithm cannot distinguish the query replied from a real world or the random oracle simulated by a function.

The properties of determinism and uniform output mean that the output of a random oracle has an entropy greater than that of its input. However by Shannon's entropy theory, a deterministic function can never amplify entropy. Therefore random oracle does not exist in the real world.

A hash function in the real world has its output values following some probability distribution which may not be discernible by a polynomially bounded distinguisher. Thus a real-world hash function only emulated behavior of a random oracle behavior to a precision where the difference is hopefully a negligible quantity. Despite of its impractical assumptions, the paradigm is useful to yield an efficient solution to prove the security of a protocol. It is better than no proof shown.

Many signature or encryption schemes used rewindings of hashings and (or) observing hashing input and output in their reductionist security proofs, like the Schnorr signature. However the result of Barak el al. [3, 4] and Goldwasser and Kalai [53] proved the insecurity of the random oracle model as it is used in the Fiat-Shamir paradigm. Several papers proved that some popular cryptosystems previously proved secure in the random oracle were actually provably insecure when the random oracle was instantiated by any real-world hashing functions [29, 5]. As a result, recently there are many new signature schemes which try to prove their security without random oracles.

Chapter 3

Literature Review

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

3.1 Identity Based Signatures

Since the introduction of identity based cryptography by Shamir [83] in 1984, there are some identity based signature schemes developed. The identity based signatures and identity based identifications are summarized by Bellare et al. [6]. They can be divided according to the underlying intractability assumptions.

- RSA: These include Shamir's first IBS scheme [83], the Guillou-Quisquater scheme [54], the Girault scheme [51], the Okamoto scheme [75], etc.

- Factorization: the Fiat-Shamir IBS scheme [45], the Ohta-Okamoto scheme [74], the Fischlin-Fischlin scheme [47], etc.

- Discrete Logarithm: the Beth scheme [10], the Bellare et al. scheme [6], etc.
- Pairing: the Sakai et al. scheme [80], the Cha-Cheon scheme [33], the Hess scheme [56], etc.

3.2 Identity Based Encryption

The first several proposals for IBE are not satisfactory [42, 86, 88, 69]. Some require that users do not collude. Some require tamper resistant hardware. Other require the TA to spend a lot of time on private key generation. Identity based encryption becomes practical only since 2001, when Boneh and Franklin [18] used a new mathematical tools called "Pairings". Since then, there are many identity based cryptosystems built using pairings. Pairings also improve many existing non-identity based cryptosystems.

Several IBE schemes [30, 12, 55] are secure without random oracles under a weaker "selective-ID" model [30]. Recently, [13] and [90] proposed IBE schemes which are provably secure without random oracles under the stronger model of [18].

Recently, there is a new extension for IBE scheme [79] which suggests the use of biometric identities, such as iris scan or fingerprint.

3.3 Identity Based Signcryption

Zheng [102] proposed that encryption and signature can be combined as "signcryption" which can be more efficient in computation than running encryption and signature separately.

There are some papers [66, 22, 63, 72, 39, 64] concerning the combination of identity based signature and encryption to form IBSC schemes. The most expensive single operation is pairing computations. Schemes of [66, 22, 64] use 5 pairings, while [63, 39] use 6, and [72] uses 4. [22] is proven secure in a stronger model than [66, 63]. [72] has no security proof.

3.4 Identity Based Blind Signatures

Blind signatures was introduced by Chaum [34]. Blind signature is described as follows: Upon request from Warden, a signing oracle makes a commitment, then blindly signs a message for Warden. Warden deblinds the signature such that the signing oracle knows neither the message nor the output signature.

Parallel one-more forgery against blind signature is that an attacker interacts with a signer l times and produces $l+1$ signatures from these interactions. Schnorr [81] reduced the parallel one-more unforgeability of the blind Schnorr signature to the ROS Problem. Some identity based blind signature schemes was proposed in [98, 99, 100].

3.5 Identity Based Group Signatures

Group signature, introduced by Chaum and van Heyst [35], allows any member of a group to sign on behalf of the group. However, the identity of the signer is kept secret. Anyone can verify that the signature is signed by a group member, but cannot tell which one. An open authority has a secret key to revoke the anonymity of any signature in case of dispute.

Identity based group signature is firstly proposed by Park et al. [76]. [68] showed that the anonymity of the scheme was not guaranteed. Tseng and Jan [87] presented a novel ID-based group scheme. However, it is universally forgeable [61] and not coalition-resistant [60]. Several identity based group signature schemes are proposed in [32], [37]. [32] requires a new pair of certificate for each signature.

3.6 Hierarchical Identity Based Cryptography

Hierarchical identity based cryptography (HIBS and HIBE) was proposed in [50] and [58] proposed another HIBE. Recently, Boneh et al. [17] (preliminary papers [31, 19]) suggested some methods to construct CCA secure ℓ-level HIBE scheme from a CPA $(\ell+1)$-level HIBE scheme. Several HIBE without random oracles are proposed in [12, 13, 90, 15] using this result. Hierarchical identity based signcryption is firstly proposed in [40].

Chapter 4

Blind Identity Based Signcryption

Blind signature was introduced by Chaum [34], which provides anonymity of users in applications such as e-cash. It allows users to get a signature of a message in a way that the signer learns neither the message nor the resulting signature.

Privacy and authenticity are also the basic aims of public key cryptography. We have encryption and signature to achieve these aims. Zheng [102] proposed that encryption and signature can be combined as "signcryption" which can be more efficient in computation than running encryption and signature separately. The security of signcryption is discussed by An et al. [1].

We present the first blind identity based signcryption (BIBSC). Roughly speaking, BIBSC works as follows: Upon request from Warden, a blind signcryption oracle makes a commitment, then blindly signs and computes the randomness term in the encryption part. Warden deblinds the signature and uses the randomness term returned to produce a signcryption.

We make the following contributions to the literature:

1. We present the first blind identity based signcryption (BIBSC).

2. We formulate the first BIBSC security models to define security notions including blindness and parallel one-more unforgeability (p1m-uf).

3. We construct the first BIBSC scheme from pairings, and prove its security. The blindness of our BIBSC scheme is statistical ZK, and the p1m-uf is reduced to Schnorr's ROS Problem in the random oracle model plus the generic group and pairing model (GGPM).

4. We introduce the generic group and pairing model (GGPM) which is an extension of the generic group model [73, 84, 81] by including support for pairings. We use this model to prove p1m-uf of our BIBSC scheme.

5. We introduce a strengthening of Boyen's [22] security model for identity based signcryption (IBSC) to add support of authenticated encryption.

6. We construct the first proven secure IBSC scheme in the strengthened model. It is also the fastest and shortest IBSC scheme in our model as well as in Boyen's [22] model.

7. The shortcomings of several existing IBSC schemes in the strengthened model are shown.

4.1 Schnorr's ROS problem

Parallel one-more forgery against blind signature is that an attacker interacts with a signer l times and produces $l + 1$ signatures from these interactions. Schnorr [81] reduced the parallel one-more unforgeability (p1m-uf) of the blind Schnorr signature to the ROS Problem in the random oracle plus generic group model (ROM+GGM). The followings are from Schnorr[81]:

Definition 28 *(ROS problem) Find an overdetermined, solvable system of linear equations modulo q with random inhomogeneities. Specifically, given an oracle random function* $F : Z_q^l \leftarrow Z_q$ *, find coefficients* $a_{k,i} \in Z_q$ *and a solvable system of* $l+1$ *distinct equations of Eq. (1) in the unknowns* $c_1, \ldots, c_l$ *over* Z_q*:*

$$a_{k,1}c_1 + \ldots + a_{k,l}c_l = F(a_{k,1}, \ldots, a_{k,l}) \textit{ for } k = 1, \ldots, t. \quad (1)$$

Theorem 1 *[81] Given generator g, public key h and an oracle for H, let a generic adversary* $\mathcal{A}$ *performs t generic steps and interacts with a signer for l times. If* $\mathcal{A}$ *succeeds in a parallel attack to produce* $l+1$ *signatures with a probability of success better than* $\binom{t}{2}/q$*, then* $\mathcal{A}$ *must solve the ROS-problem in ROM+GGM.*

4.2 BIBSC and Enhanced IBSC Security Model

We define the first security model for BIBSC and also an enhancement of Boyen's security model for IBSC. For logistics, we present the latter first.

Intuitions: Basically, signcryption reuses the randomness in signing as the randomness in encryption, to achieve bandwidth conservation. Lower complexity is also a goal. In blind signcryption, below, the "prover oracle" delivers both the blind signature as well as the intermediate encryption results which reuses the randomness. In comparison, the prover oracle in a blind signature scheme delivers only the signature.

In the naive sign-then-encrypt (StE) instantiation, the recipient can decrypt, and then re-encrypt the (sender) signed plaintext to a third party. The resulting signcryption is a valid signcryption but the signer and the encryptor are distinct. Boyen's *ciphertext unlinkability* [22] extends this basic idea. In the naive encrypt-then-sign (EtS) instantiation, the encryptor and the signer are assured to be the same. The *authenticated encryption* [1] extends this basic idea. Our security model supports both ciphertext unlinkability and authenticated encryption in two different but closely related *dual versions.*

4.2.1 Enhanced IBSC Security Model

We present an enhancement of Boyen's security model for IBSC. The main addition is to add support for *authenticated encryption.* The signer cannot deny signcrypting the message to the recipient. Boyen's model is restricted to *ciphertext unlinkability* where this assurance is not required. Our model below is capable of supporting authenticated encryption and ciphertext unlinkability.

4.2.1.1 Primitives

An IBSC scheme consists of four algorithms: (Setup, Extract, Signcrypt, Unsigncrypt). The algorithms are specified as follows:

Setup: On input a security parameter k, the TA generates $\langle \zeta, \pi \rangle$ where ζ is the randomly generated master key, and π is the corresponding public parameter.

Extract: On input ID, the TA computes its corresponding private key S_{ID} (corresponding to $\langle \zeta, \pi \rangle$) and sends back to its owner in a secure channel.

Signcrypt: On input the private key of sender A, S_A, recipient identity ID_B and a message m, outputs a ciphertext σ corresponding to π.

Unsigncrypt: On input private key of recipient B, S_B, and ciphertext σ, decrypt to get sender identity ID_A, message m and signature s corresponding to π. Verify s and verify if encryptor = signer. Output $\top$ for "true" or $\bot$ for "false".

We make the consistency constraint that if $\sigma \leftarrow \mathit{Signcrypt}(S_A, ID_B, m)$, then $m \leftarrow \mathit{Unsigncrypt}(S_B, \sigma)$.

4.2.1.2 Indistinguishability

Indistinguishability for IBSC against adaptive chosen ciphertext attack (IND-IBSC-CCA2) is defined as in the following game. The adversary is allowed to query the random oracles, key extraction oracle, signcryption oracle and unsigncryption oracle. The game is defined as follows:

1. The simulator selects the public parameter and sends to the adversary.
2. The adversary performs polynomial number of oracle queries adaptively.
3. The adversary generates m_1, ID_{A1}, ID_{B1}, and sends to the simulator. The adversary knows S_{A1}. The simulator generates m_0, ID_{A0}, ID_{B0}, randomly chooses $b \in_R \{0, 1\}$. The simulator delivers the challenge ciphertext $\sigma \leftarrow \mathit{Signcrypt}(S_{Ab}, ID_{Bb}, m_b)$ to the adversary.
4. The adversary performs polynomial number of oracle queries adaptively.
5. The adversary tries to compute b, in the following three sub-games:
   (a) The simulator ensures $B0 = B1$, $m_0 = m_1$.
   (b) The simulator ensures $A0 = A1$, $m_0 = m_1$.
   (c) The simulator ensures $A0 = A1$, $B0 = B1$.

The adversary wins the game if he can guess b correctly. The *advantage* of the adversary is the probability, over half, that he can compute b accurately.

The oracles are defined as follows:

Key extraction oracle $\mathcal{KEO}$: Upon input an identity, the key extraction oracle outputs the private key corresponding to this identity.

Signcryption oracle $\mathcal{SO}$: Upon input m, ID_A, ID_B, the signcryption oracle produces a valid signcryption σ for the triple of input.

Unsigncryption oracle $\mathcal{UO}$: Upon input ciphertext σ and recipient ID, the unsigncryption oracle outputs the decryption result and the verification outcome.

Oracle query to $\mathcal{KEO}$ with input ID_{B0} or ID_{B1} is not allowed. Oracle query to $\mathcal{SO}$ with input (m_1, ID_{A1}, ID_{B1}) is not allowed. [1] Oracle query to $\mathcal{UO}$ for the challenge ciphertext from the simulator is not allowed.

Definition 29 *(Indistinguishability) An IBSC is* IND-IBSC-CCA2 *secure if no PPT adversary has a non-negligible advantage in any of the sub-games above.*

Our security notion above is a strong one. It incorporates previous security notions including *insider-security* in [1], *indistinguishability* in [66], and *anonymity* in [22].

4.2.1.3 Existential unforgeability

Existential unforgeability against adaptive chosen message attack for IBSC (EU-IBSC-CMA) is defined as in the following game. The adversary is allowed to query the random oracles, $\mathcal{KEO}$, $\mathcal{SO}$ and $\mathcal{UO}$, which are defined in the above section. The game is defined as follows:

1. The simulator selects the public parameter and sends to the adversary.

2. The adversary performs polynomially number of oracle queries adaptively.

[1] We use a weaker model here for our construction. In a stronger model, this restriction for $\mathcal{SO}$ is not needed.

3. The adversary delivers a recipient identity ID_B and a ciphertext σ.

The adversary wins the game if he can produce a valid (σ, ID_B) such that σ can be decrypted, under the private key of ID_B, to a message m, sender identity ID_A and a signature s which passes all verification tests.

Oracle query to $\mathcal{KEO}$ with input ID_A is not allowed. The adversary's answer (σ, ID_B) should not be computed by $\mathcal{SO}$ before.

Definition 30 *(Existential Unforgeability) An IBSC is* EU-IBSC-CMA *secure if no PPT adversary has a non-negligible probability in winning the game above.*

The adversary is allowed to get the private key of the recipient in the adversary's answer. This gives us an *insider-security* in [1]. This model for *authenticated encryption* is stronger than Boyen's [22] existential unforgeability in the sense that our model provides non-repudiation for the ciphertext while Boyen's provides non-repudiation for the decrypted signature only. For *ciphertext unlinkability*, we have to add one more restriction for our model. Oracle queries to $\mathcal{SO}$ for (ID_A, m) in the adversary's answer using any recipient identity are not allowed. Then the model changes to non-repudiation for signature only.

4.2.2 BIBSC Security Model

We will propose the primitives of blind version of IBSC and then define the security notions for blindness and parallel one-more unforgeability.

4.2.2.1 Primitives

A BIBSC is a five-tuple (Setup, Extract, BlindSigncrypt, Warden, Unsigncrypt) where Setup, Extract and Unsigncrypt are identical as primitives in IBSC.

(BlindSigncrypt, Warden) is a 3-move interactive protocol as follows. Input to BlindSigncrypt is the sender identity ID_A and its private key S_A, and the recipient identity ID_B. Input to Warden is ID_A, ID_B and a message m.

1. BlindSigncrypt sends a commitment X to Warden.
2. Warden challenges BlindSigncrypt with h.
3. BlindSigncrypt sends back the response W and V to Warden.

Finally Warden outputs a ciphertext σ.

4.2.2.2 Blindness

Here we define the blindness of BIBSC. The adversary is allowed to make q_B queries to blind signcryption oracle $\mathcal{BSO}$, q_H queries to random oracles, q_K queries to $\mathcal{KEO}$, q_S queries to $\mathcal{SO}$, and q_U queries to $\mathcal{UO}$. The adversary keeps the transcript $\mathcal{T}$ recording the interaction between BlindSigncrypt and Warden.

Definition 31 *(Blindness) A BIBSC is blind if given a ciphertext σ by Warden, $Prob\{\sigma \ by \ Warden\} = Prob\{\sigma \ by \ Warden|\mathcal{T}\}$*

4.2.2.3 Parallel One-more Unforgeability

Parallel one-more unforgeability for BIBSC (p1m-uf) is defined as in the following game. It is similar to the one-more forgery for traditional blind signature schemes [7, 11, 100].

1. The adversary gives the sender identity ID_A to the simulator.
2. The adversary makes a total of q_B queries to blind signcryption oracle $\mathcal{BSO}_{ID_k}$, $1 \leq k \leq K$, q_H queries to random oracles, q_K queries to $\mathcal{KEO}$, q_S queries to $\mathcal{SO}$, and q_U queries to $\mathcal{UO}$.

3. The adversary delivers $q_B + q_S + 1$ tuples (ID_i, σ_i) to the simulator, $1 \leq i \leq q_B + q_S + 1$.

The adversary wins the game if he can produce $q_B + q_S + 1$ valid distinct tuples (ID_i, σ_i) that can decrypts, under the private key of ID_i, to message m_i, sender identity ID_A, and signature s_i which passes the verification tests. The $\mathcal{SO}$, $\mathcal{UO}$ and $\mathcal{KEO}$ are same as the one in IBSC. We have the new interactive $\mathcal{BSO}$:

$\mathcal{BSO}_{ID_A}$: Upon input ID_B, it returns a number X. Then input a number h. It produces an output (W, V) based on sender ID_A, recipient ID_B, X and h.

It is required that the private key of ID_A is never extracted by $\mathcal{KEO}$.

Definition 32 *(Parallel One-more Unforgeability) A BIBSC is* p1m-uf *secure if no PPT adversary has a non-negligible probability in winning the above game.*

4.3 Efficient and Secure BIBSC and IBSC Schemes

We present our constructions of efficient and secure BIBSC and IBSC schemes from pairings. For logistics of presentation, we present the IBSC scheme first.

4.3.1 Efficient and Secure IBSC Scheme

This IBSC scheme follows the primitives in Section 2. Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be (multiplicative) cyclic groups of order p. The pairing is given as $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Now we define our scheme as follows.

Setup: The setup of TA is similar to [18]. On input a security parameter $\lambda \in \mathbb{N}$, a generator $G[1^\lambda]$ generates $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$, p and $\hat{e}$. The TA chooses a generator $P \in \mathbb{G}_1$ and picks a random $s \in \mathbb{Z}_p$ as the master key. Then the TA sets $P_{TA} = P^s \in \mathbb{G}_1$. After that the TA chooses cryptographic hash functions $H_0 : \{0,1\}^* \rightarrow \mathbb{G}_2, H_1 : \{0,1\}^* \times \mathbb{G}_2 \times \{0,1\}^* \rightarrow \mathbb{Z}_p, H_2 : \mathbb{G}_T \rightarrow \{0,1\}^*, H_3 : \mathbb{G}_T \times \{0,1\}^* \rightarrow \mathbb{G}_2$. The system parameters are $\langle p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, P, P_{TA}, H_0, H_1, H_2, H_3 \rangle$.

Extract: Given a user identity string $ID \in \{0,1\}^*$, his public key is $Q_{ID} = H_0(ID) \in \mathbb{G}_2$. His private key $S_{ID} = (Q_{ID})^s \in \mathbb{G}_2$ is calculated by TA.

Signcrypt: Suppose Alice wants to signcrypt a message m to Bob. Assume Alice's identity is ID_A with public key Q_A and private key S_A. Bob's identity is ID_B.

- Sign: Alice chooses a random $r \in \mathbb{Z}_p$ and computes:

$$\begin{aligned} X &= P^r \in \mathbb{G}_1 \\ h &= H_1(m, X, ID_B) \in \mathbb{Z}_p \\ W &= {S_A}^h {Q_A}^r \in \mathbb{G}_2 \end{aligned}$$

- Encrypt: Alice computes $Q_B = H_0(ID_B) \in \mathbb{G}_2$ and:

$$\begin{aligned} V &= \hat{e}({P_{TA}}^r, Q_B) \in \mathbb{G}_T \\ Y &= H_3(V, ID_A) \oplus W \in \mathbb{G}_2 \\ Z &= H_2(V) \oplus \langle ID_A, m \rangle \in \{0,1\}^* \end{aligned}$$

  Alice outputs a ciphertext $\sigma = \langle X, Y, Z \rangle$ and sends to Bob.

Unsigncrypt: Bob receives the ciphertext $\sigma = \langle X, Y, Z \rangle$.

- Decrypt: Assume the private key of Bob is S_B. Bob computes:

$$\begin{aligned} V' &= \hat{e}(X, S_B) \\ \langle ID_A, m \rangle &= H_2(V') \oplus Z \end{aligned}$$

Output $\langle ID_A, m\rangle$ together with $\langle X, Y, V'\rangle$ to Verify.

- Verify: Bob computes $W' = H_3(V', ID_A) \oplus Y$ and compares if:

$$\hat{e}(P, W') = \hat{e}(XP_{TA}{}^h, Q_A) \text{ where } h = H_1(m, X, ID_B)$$

Output $\top$ if the above verification is true, or output $\perp$ if false.

In Section 4.2.1, Unsigncrypt also requires the verification for checking encryptor = signer. It is implicitly done in Decrypt and Verify as both of them use the same X in σ to decrypt and verify.

Finally, we show the consistency constraint is satisfied in Decrypt and Verify. In Decrypt, V can be recovered as: $\hat{e}(X, S_B) = \hat{e}(P^r, Q_B{}^s) = \hat{e}(P_{TA}{}^r, Q_B)$. In Verify, if the signature is valid, both sides should be equivalent because: $\hat{e}(P, W) = \hat{e}(P, S_A{}^h Q_A{}^r) = \hat{e}(P, Q_A{}^{(sh+r)}) = \hat{e}(P^{(r+sh)}, Q_A) = \hat{e}(XP_{TA}{}^h, Q_A)$.

Theorem 2 *Our IBSC scheme is* IND-IBSC-CCA2 *secure provided the co-BDH Problem is hard in the random oracle model.*

Proof. Setting up: Simulator $\mathcal{S}$ is given $(P, P^\alpha, P^\beta, Q)$ and wants to compute $\hat{e}(P, Q)^{\alpha\beta}$. $\mathcal{S}$ sends the system parameter to Adversary $\mathcal{A}$ with $P_{TA} = P^\beta$ as in Setup. $\mathcal{S}$ randomly picks η_Q from $\{1, 2, ..., \mu_0\}$, where μ_0 is the number of queries to H_0.

Simulating Oracles: As regards queries to the oracles:

- Query on H_0 for identity ID is handled as follows:
  - The η_Q-th distinct query to H_0 is back patched to the value Q. The corresponding identity is denoted as ID_Q. Adds the entry $\langle ID_Q, Q\rangle$ to tape L_0, and returns the public key Q.

  - Otherwise, picks a random $\lambda_i \in F_p^*$, adds the entry $\langle ID, \lambda_i \rangle$ to the tape L_0, and return the public key $Q_{ID} = P^{\lambda_i}$.

- Queries on H_1, H_2 and H_3 are handled by producing a random element from the codomain, and adding both query and answer to tape L_1, L_2 and L_3.

- $\mathcal{KEO}$: For input identity ID_A.
  - If $ID_A = ID_Q$, then $\mathcal{S}$ terminates its interaction with $\mathcal{A}$, having failed to guess the targeted recipient among those in L_0.
  - Otherwise, S retrieves $\langle ID_A,\ \lambda_A \rangle$ from L_0 and returns $S_A = (P^{\beta})^{\lambda_A}$.

- $\mathcal{SO}$: For input message m, sender ID_A, and recipient ID_B.
  - If $ID_A = ID_Q$, then $\mathcal{S}$ randomly chooses $r, h \in F_p^*$, and lets $X = P^r (P^{\beta})^{-h}$, $W = (Q)^r$. Then, $\mathcal{S}$ adds the tuple $\langle m, X, h \oplus ID_B \rangle$ to L_1 to force the random oracle $H_1(m, X) = h \oplus ID_B$. Finally, $\mathcal{S}$ uses $\langle X, W, m, r, ID_B \rangle$ to run Signcrypt to produce the desired ciphertext σ.
  - Otherwise, $\mathcal{S}$ retrieves $\langle ID_A, \lambda_A \rangle$ from L_0 and computes $S_A = (P^{\beta})^{\lambda_A}$. Then $\mathcal{S}$ will run Signcrypt using S_A and get ciphertext σ.

- $\mathcal{UO}$: For input recipient ID_B and ciphertext $\sigma = \langle X, Y, Z \rangle$.
  - If $ID_B = ID_Q$, then $\mathcal{S}$ searches all combinations $\langle ID_A, m, X, W \rangle$ such that $\langle m, X, h_1 \rangle \in L_1$, $\langle V, h_2 \rangle \in L_2$, $\langle V, ID_A, h_3 \rangle \in L_3$, for some h_1, h_2, h_3, V, under the constraints that $h_3 \oplus Y = W$, $h_2 \oplus Z = \langle ID_A, m \rangle$ and $\mathsf{Verify}[ID_A, m, X, W, ID_B] = \top$. Pick a $\langle ID_A, m \rangle$ in one of the combinations above to return as answer.

If no such tuple is found, the oracle signals that the ciphertext is invalid.

- Otherwise, $\mathcal{S}$ retrieves $\langle ID_B, \lambda_B \rangle$ from L_0 and computes $S_B = (P^\beta)^{\lambda_B}$. Then $\mathcal{S}$ will run Unsigncrypt using S_B to get $\langle ID_A, m \rangle$ or $\perp$.

Extraction: As in the IND-IBSC-CCA2 game, at some point $\mathcal{A}$ chooses plaintext m_1, sender ID_{A1}, and recipient ID_{B1} on which he wishes to be challenged. $\mathcal{S}$ responds with challenge ciphertext $\langle X, Y, Z \rangle$, where $X = P^\alpha$. Y and Z are random strings of appropriate size. All further queries by $\mathcal{A}$ are processed adaptively as in the oracles above.

Finally, $\mathcal{A}$ returns its final guess. $\mathcal{S}$ ignores the answer from $\mathcal{A}$, randomly picks an entry $\langle V, h_2 \rangle$ in L_2, and returns V as the solution to the co-BDH problem.

If the recipient identity $ID_{A1} = ID_Q$ selected by $\mathcal{S}$, to recognize the challenge ciphertext $\langle X, Y, Z \rangle$ with $X = P^\alpha$ is incorrect, $\mathcal{A}$ needs to query random oracle $H_2(V)$ with $V = \hat{e}(X, S_Q) = \hat{e}(P^\alpha, Q^\beta) = \hat{e}(P, Q)^{\alpha\beta}$. It will leave an entry $\langle V, h_2 \rangle$ on L_2, from which $\mathcal{S}$ can extract $V = \hat{e}(P, Q)^{\alpha\beta}$.

Theorem 3 *Our IBSC scheme is* EU-IBSC-CMA *secure provided the co-CDH Problem is hard in the random oracle model.*

Proof. *Setting up:* Simulator $\mathcal{S}$ is given (P, P^β, Q) and wants to compute Q^β. Others are same as in the proof sketch of Theorem 2.

Oracle Simulation: The signcryption oracle, unsigncryption oracle, and key extraction oracle are simulated in the same way as in the proof of Theorem 2.

Extraction: Assume $\mathcal{A}$ is a PPT adversary. Rewind $\mathcal{A}$ to the random oracle query whose output appears in verification of Unsigncrypt. Then we obtain $W = S_A^h Q_A^r$ and $W' = S_A^{h'} Q_A^r$ in respective forks. Combining, we can compute the co-CDH Problem if $Q_A = Q$. Then $Q^\beta = S_A = (W'/W)^{(h'-h)^{-1}}$.

Dual support of ciphertext unlinkability (CU) and authenticated encryption (AE):

One of the main difference between our IBSC scheme and Boyen's scheme [22] is that our scheme has linkability (AE) while Boyen's scheme has unlinkability (CU). In our original AE version, we include the recipient identity in the signature, such that the adversary cannot reuse the signature s by sender ID_A for other recipients and encrypt s to forge a signcryption from ID_A to the adversary himself.

As unlinkability may also be important in some applications, we provide the CU version of our scheme. The only change is that in Sign change $h = H_1(m, X)$. Other steps remain unchanged. Therefore this CU version is as efficient as the original AE version. Notice that by changing to CU, unforgeability for ciphertext reduces to unforgeability for signature only, as in [22].

4.3.2 The First BIBSC Scheme

In this BIBSC scheme, Setup, Extract and Unsigncrypt are the same as Section 4.3.1. We describe the interactive protocol for BlindSigncrypt and Warden below:

BlindSigncrypt		Warden
choose $r \in \mathbb{Z}_p$		choose $\alpha, \beta \in \mathbb{Z}_p$
send $X = P^r$	$\longrightarrow$	
		compute $\hat{X} = X^\alpha P^\beta$,
		compute $\hat{h} = H(m, \hat{X}, ID_B)$
	$\longleftarrow$	send $h = \alpha^{-1}\hat{h}$
send $W = {S_A}^h {Q_A}^r$		
and $V = \hat{e}({P_{TA}}^r, Q_B)$	$\longrightarrow$	
		compute $\hat{W} = W^\alpha {Q_A}^\beta$
		compute $\hat{V} = V^\alpha \hat{e}({P_{TA}}^\beta, Q_B)$
		compute $\hat{Y} = H_3(\hat{V}, ID_A) \oplus \hat{W}$
		compute $\hat{Z} = H_2(\hat{V}) \oplus \langle ID_A, m \rangle$
		output $\sigma = \langle \hat{X}, \hat{Y}, \hat{Z} \rangle$

Consistency is verified as:

$$\begin{aligned} \hat{e}(P, \hat{W}) &= \hat{e}(P, W^\alpha {Q_A}^\beta) & \text{and} \quad \hat{V} &= V^\alpha \hat{e}({P_{TA}}^\beta, Q_B) \\ &= \hat{e}(P, Q_A)^{s\hat{h}+\alpha r+\beta} & &= \hat{e}(P^{s(r\alpha+\beta)}, Q_B) \\ &= \hat{e}({P_{TA}}^{\hat{h}} X^\alpha P^\beta, Q_A) & &= \hat{e}(X^\alpha P^\beta, S_B) \\ &= \hat{e}(\hat{X} {P_{TA}}^{\hat{h}}, Q_A) & &= \hat{e}(\hat{X}, S_B) \end{aligned}$$

Theorem 4 *Our BIBSC scheme has blindness.*

Proof. To prove the blindness of BIBSC scheme, we show that given a valid ciphertext $\langle \hat{X}, \hat{Y}, \hat{Z} \rangle$ and any transcript of blind signcryption (X, h, W, V), there always exists a unique pair of blinding factors $\alpha, \beta \in \mathbb{Z}_p^*$. Since the blinding factors are randomly chosen, the blindness of BIBSC scheme is achieved.

Given a valid ciphertext $\langle \hat{X}, \hat{Y}, \hat{Z} \rangle$, then there exists a unique $(\hat{X}, \hat{W}, \hat{V}, m)$ for this ciphertext. Then for any transcript of blind signcryption (X, h, W, V), the following equations must hold for $\alpha, \beta \in \mathbb{Z}_p^*$:

$$\begin{aligned} \hat{X} &= X^\alpha P^\beta \\ h &= \alpha^{-1} H_1(m, \hat{X}) \\ \hat{W} &= W^\alpha {Q_A}^\beta \\ \hat{V} &= V^\alpha \hat{e}({P_{TA}}^\beta, Q_B) \end{aligned}$$

From the second equation, we see that there exists a blinding factor $\alpha = H_1(m, \hat{X})/h$. For this α, there exists a blinding factor β from the first equation and $\beta = log_P(\hat{X}X^{-\alpha})$. Therefore we have to show that these blinding factors α, β satisfy the last two equations.

Notice that there exists a S_B which is the private key for Q_B. Then:

$$\begin{aligned} \hat{V} &= \hat{e}(\hat{X}, S_B) \\ &= \hat{e}(X^\alpha P^\beta, S_B) \\ &= \hat{e}(X, S_B)^\alpha \hat{e}(P^\beta, S_B) \\ &= V^\alpha \hat{e}({P_{TA}}^\beta, Q_B) \end{aligned}$$

Furthermore, $\langle \hat{X}, \hat{W}, m \rangle$ is a valid signature. Therefore we have:

$$\begin{aligned} \hat{e}(P, \hat{W}) &= \hat{e}(\hat{X}, Q_A)\hat{e}(P_{TA}, Q_A)^{H_1(m,\hat{X})} \\ &= \hat{e}(X^\alpha P^\beta, Q_A)\hat{e}(P_{TA}, Q_A)^{\alpha h} \\ &= \hat{e}(X{P_{TA}}^h, Q_A)^\alpha \hat{e}(P^\beta, Q_A) \\ &= \hat{e}(P, W)^\alpha \hat{e}(P, {Q_A}^\beta) \\ &= \hat{e}(P, W^\alpha {Q_A}^\beta) \end{aligned}$$

Hence, given a valid ciphertext $\langle \hat{X}, \hat{Y}, \hat{Z} \rangle$ and any transcript of blind signcryption (X, h, W, V), there always exists a unique pair of blinding factors $\alpha, \beta \in \mathbb{Z}_p^*$. Therefore, $\Pr[\sigma \text{ by Warden}] = \Pr[\sigma \text{ by Warden}|\mathcal{T}]$. The blindness of BIBSC scheme is proved.

Theorem 5 *Our BIBSC scheme is* p1m-uf *secure provided Schnorr's ROS Problem is hard in the ROM+GGPM.*

Proof. For simplicity, assume that $\mathcal{A}$ makes all $\mathcal{SO}$ queries for challenge identity ID_A. Otherwise, $\mathcal{A}$ wins with a smaller probability. This proof refers to a generic adversary $\mathcal{A}$ performing some t generic steps, including some q_B interactions $(X_1,h_1,W_1,V_1),\cdots,(X_{q_B},h_{q_B},W_{q_B},V_{q_B})$ with $\mathcal{BSO}$, some q_S computations $(X_{q_B+1},h_{q_B+1},W_{q_B+1},V_{q_B+1}),\cdots,(X_{q_B+q_S},h_{q_B+q_S},W_{q_B+q_S},V_{q_B+q_S})$ with $\mathcal{SO}$, producing some $t'^{(u)}$ group elements in G_u. We let $\vec{r} = (r_1, \cdots, r_{q_B+q_S})$

denote $\mathcal{BSO}$ or $\mathcal{SO}$ random coins. Let $f_1 = P, f_2 = P_{TA}, f_3, \cdots, f_{t'(1)} \in \mathbb{G}_1$ denote the group elements of $\mathcal{A}$'s computation. The generic $\mathcal{A}$ computes $f_j = P^{a_{j,-1}} P_{TA}^{a_{j,0}} \prod_{\ell=1}^{q_B+q_S} X_\ell{}^{a_{j,\ell}}$ where X_ℓ are $\mathcal{BSO}$ commitments and the exponents depend arbitrarily on previously computed non-group data.

Schnorr's Lemma 2 implies DLP is hard (uncomputable by PPT generic adversary) in GGM. Similarly, it applies here. It is hard to get s from ${Q_B}^s$.

Let $\mathcal{A}$'s outputs $(\hat{X}_i, \hat{W}_i, \hat{V}_i)$ be valid for message $\hat{m}_i$, sender ID_A and recipient ID_{B_i}, $1 \leq i \leq q_B + q_S + 1$. Then we have $\hat{h}_i = H_1(\hat{X}_i, \hat{m}_i, ID_{B_i})$ for some hash query satisfying $\hat{e}(\hat{X}_i P_{TA}^{\hat{h}_i}, Q_A) = \hat{e}(P, \hat{W}_i)$. Let $\hat{X}_i = f_{\sigma_i}^{(1)}$.

The equation $\hat{e}(P, \hat{W}_i)\hat{e}(P_{TA}^{-\hat{h}_i}, Q_A)$=$\hat{e}(f_{\sigma_i}, Q_A)$=$\hat{e}(P^{a_{\sigma_i,-1}} P_{TA}^{a_{\sigma_i,0}} \prod_{\ell=1}^{q_B+q_S} X_\ell{}^{a_{\sigma_i,\ell}}$, $Q_A)$ and $\hat{e}(X_\ell, Q_A)$=$\hat{e}(P,\, W_\ell)\hat{e}(P_{TA}^{-h_\ell},\, Q_A)$ imply:

$$\hat{W}_i \;=\; {Q_A}^{a_{\sigma_i,-1}} \cdot \prod_{\ell=1}^{q_B+q_S} W_\ell^{a_{\sigma_i,\ell}} \cdot {Q_A}^{(a_{\sigma_i,0} - \sum_{\ell=1}^{q_B+q_S} a_{\sigma_i,\ell} h_\ell + \hat{h}_i)s}$$

If $\hat{h}_i = -a_{\sigma_i,0} + \sum_{\ell=1}^{l} a_{\sigma_i,\ell} h_\ell$, then $\mathcal{A}$ can easily compute the correct $\hat{W}_i$. Then we have $\hat{W}_i = {Q_A}^{a_{\sigma_i,-1}} \prod_{\ell=1}^{l} W_\ell^{a_{\sigma_i,\ell}}$ where $W_1,\cdots,W_l$, $a_{\sigma_i,-1},\cdots,a_{\sigma_i,l}$ are known to $\mathcal{A}$.

Conversely, $\mathcal{A}$ must select $h_1, \cdots, h_l$ as to zero the coefficient involving the master secret key s. Otherwise we can recover ${Q_A}^s$ from $W_1,\cdots,W_l$, $a_{\sigma_i,-1},\cdots,a_{\sigma_i,l}$, $\hat{h}_i$, $\hat{W}_i$ which are known to $\mathcal{A}$. Then it can solve the 1m-co-CDH problem, as we get q_K private keys from $\mathcal{KEO}$. The probability of solving 1m-co-CDH in GGPM is negligible. Hence $\mathcal{A}$ must solve the ROS problem.

Remark: In our proof, we use an alternative representation for $\hat{Y}$ and $\hat{Z}$. Let θ_4 (resp. θ_5) be a bijective mapping from $\mathbb{G}_2$ to $\mathbb{G}_4$ (resp. from $\{0,1\}^*$ to $\mathbb{G}_5$) where $\mathbb{G}_4$ (resp. $\mathbb{G}_5$) is a cyclic group. Change $H_2 : \mathbb{G}_T \rightarrow G_5, H_3 : \mathbb{G}_T \times \{0,1\}^* \rightarrow \mathbb{G}_4$. Then $\hat{Y} = H_3(\hat{V}, ID_A) \oplus \theta_4(\hat{W}) \in \mathbb{G}_4$ and

$\hat{Z} = H_2(\hat{V}) \oplus \theta_5(\langle ID_A, m\rangle) \in \mathbb{G}_5$. In Unsigncrypt, we can use θ_4^{-1} and θ_5^{-1} to recover the message. The efficiency and security of our BIBSC scheme will not be affected.

4.4 Generic Group and Pairing Model

We briefly introduce the generic group and pairing model (GGPM) by extending the generic group model (GGM) of [73, 84, 81], to include the support for the pairing oracle. There are two types of data, namely, group elements in $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$, and non-group data. The group cardinalities are prime numbers p_1, p_2, p_3 respectively, with $p_1 = p_2 = p_3 = p$. Non-group data are integers in $\mathbb{Z}$ (or in $\mathbb{Z}_p$ depending on convention). The *base elements* of $\mathbb{G}_T$ can be randomly generated, obtained from the blind signcryption oracle, or computed as the pairing of one element from $\mathbb{G}_1$ and one element from $\mathbb{G}_2$. The GGPM consists of:

1. Three GGMs, one for each of $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$. Denote their *encodings* by $\theta_i : G_i \to S_i$, $i = 1, 2, 3$.

2. A pairing oracle, $\hat{e} : S_1 \times S_2 \to S_3$, satisfying bilinear properties.

3. Other oracles in the security model such as $\mathcal{BSO}$, $\mathcal{KEO}$ and random oracle.

The encodings θ_i are that non-group operations are meaningless. Similar to [81] each *generic step* is a computation of one of the following:

mex-1: $Z_q^{d_1} \times \mathbb{G}_1^{d_1} \to \mathbb{G}_1, (a_1^{(1)}, \cdots, a_{d_1}^{(1)}, g_1^{(1)}, \cdots, g_{d_1}^{(1)}) \mapsto \prod_i (g_i^{(1)})^{a_i^{(1)}}$
mex-2: $Z_q^{d_2} \times \mathbb{G}_2^{d_2} \to \mathbb{G}_2, (a_1^{(2)}, \cdots, a_{d_2}^{(2)}, g_1^{(2)}, \cdots, g_{d_2}^{(2)}) \mapsto \prod_{i'} (g_{i'}^{(2)})^{a_{i'}^{(2)}}$
mex-3: $\mathbb{Z}_p^{d_3+d_1d_2} \times \mathbb{G}_T^{d_3} \times \mathbb{G}_1^{d_1}\mathbb{G}_2^{d_2} \to \mathbb{G}_T,$
$(a_1^{(3)}, \cdots, a_{d_3+d_1d_2}^{(3)}, g_1^{(3)}, \cdots, g_{d_3}^{(3)}, (g_1^{(1)}, g_1^{(2)}), \cdots, (g_{d_1}^{(1)}, g_{d_2}^{(2)}))$

$\mapsto \prod_{i=1}^{d_3}(g_i^{(3)})^{a_i^{(3)}} \prod_{j=1}^{d_1}\prod_{k=1}^{d_2} \hat{e}(g_j^{(1)}, g_k^{(2)})^{a^{(3)}_{d_3+d_2(j-1)+k}}$

$\mathsf{mex\text{-}p} : \mathbb{Z}_p^{d_1+d_2} \times \mathbb{G}_1^{d_1} \times \mathbb{G}_2^{d_2} \to \mathbb{G}_T,$

$(a_1^{(4)}, \cdots, a_{d_1}^{(4)}, a_1^{(5)}, \cdots, a_{d_2}^{(5)}, g_1^{(1)}, \cdots, g_{d_1}^{(1)}, g_1^{(2)}, \cdots, g_{d_2}^{(2)})$

$\mapsto \prod_j \prod_k \hat{e}(g_j^{(1)}, g_k^{(2)})^{a_j^{(4)} a_k^{(5)}}$

The elements $g_i^{(1)}$'s are P, P_{TA}, $\mathcal{BSO}$ commitments X_i's, and randomly generate $\mathbb{G}_1$ elements. The elements $g_i^{(2)}$'s are Q_{ID}'s, S_{ID}'s, $\mathcal{BSO}$ responses W_i's, and randomly generate $\mathbb{G}_2$ elements. The elements $g_i^{(3)}$'s are $\mathcal{BSO}$ responses V_i's, randomly generate $\mathbb{G}_T$ elements, and pairing oracle outputs. Similar to [81], we can omit randomly generated group elements, below, w.l.o.g.

A (non-interactive) **generic algorithm** is a sequence of t_{total} generic steps

1. Inputs are: $f_1^{(u)}, \cdots, f_{t'_u}^{(u)} \in G_u$ for $u = 1, 2, 3$, $1 \le t'_u < t_{total}$, where $t' = \sum_u t'_u < t_{total}$ and non-group data like $\mathbb{Z}_p$ in given ciphertext or signature.

2. Computation steps are: $f_i^{(u)} = \prod_{j=1}^{i-1}(f_j^{(u)})^{a_{i,j}^{(u)}}$, for $i = t'_u + 1, \cdots, t_u$, $u = 1, 2$, and $f_i^{(3)} = \prod_{j=1}^{i-1}(f_j^{(3)})^{a_{i,j}^{(3)}} \cdot \prod_{1 \le k,\ell < t} \hat{e}(f_k^{(1)}, f_\ell^{(2)})^{b_{i,k,\ell}}$ for $i = t'_3 + 1, \cdots, t_3$, where $t_{total} = t_1 + t_2 + t_3 + t_4$ and exponents $a_{i,j}^{(u)}$ depends arbitrarily on i, j, and non-group inputs.

3. Outputs are: non-group data and group elements $f_{\sigma_1}^{(u)}, \cdots, f_{\sigma_d}^{(u)}$where the integers $\sigma_1, \cdots, \sigma_d \in \{1, \cdots, t_u\}$ that depend arbitrarily on the non-group input.

The generic adversary can also perform equality test, if-then-else, looping, and other logical operations. We omit discussions about them here.

In the generic algorithm, each computation step $f_\sigma^{(u)}$ must be represented as the product of powers of group elements $g_i^{(1)}$'s, $g_{i'}^{(2)}$'s, $g_{i''}^{(3)}$'s, and

$\hat{e}(g_k^{(1)}, g_\ell^{(2)})$'s. There are only polynomially many group elements involved in any PPT algorithm. Each step can be represented as a sequence of exponents, and that representation should be unique. A *collision* is when a step can have multiple representations w.r.t. the bases consisting of the prescribed set of group elements. The following lemma shows the *collision* probability for $f_i^{(1)}, f_j^{(2)}, f_k^{(3)}$ are negligible except when involving oracle queries. The proof is similar to Schnorr's Lemma 1 and omitted.

Lemma 6 *In an arbitrary instantiation of the generic groups and the generic pairing, the probability of a PPT generic algorithm being able to compute a collision is negligible, except the collisions obtained via oracle queries. The probability is taken over randomized instantiations of all randomly generated base elements.*

Oracle assisted collisions are obtained from the $\mathcal{BSO}$ which are of the type $\hat{e}(A, B) = \hat{e}(C, D)$ in $\mathbb{G}_T$. The $\mathcal{KEO}$ also yields collisions in $\mathbb{G}_2$. The identity based characteristics need special attention in the proof of this lemma.

Next we elaborate on **interactive generic algorithms**. We count the following generic steps:

- group operations mex-1, mex-2, mex-3, mex-p
- queries to hash oracle H
- queries to key extraction oracle $\mathcal{KEO}$
- interactions with a blind signcryption oracle $\mathcal{BSO}$.

A **generic adversary** is an interactive algorithm that interacts with $\mathcal{BSO}$. The construction is similar to Schnorr's, unless specified below. The *input* consists of generators $g^{(1)}, g^{(2)}, g^{(3)}$, public keys $Q_1, \cdots, Q_K \in \mathbb{G}_2$, master public key $P_{TA} \in \mathbb{G}_1$, group order p, pairing $\hat{e}(\cdot, \cdot)$ and collection of messages, ciphertexts and so on, which can be broken into group elements and non-group data.

$\mathcal{A}$'s *transmission* to $\mathcal{KEO}$ depends arbitrarily on given group elements and non-group data. Notice that key extraction for sender's private key is not allowed.

The *restriction* is that $\mathcal{A}$ can use group elements only for generic group operations, equality tests and for queries to hash oracle and $\mathcal{KEO}$, whereas non-group data can be arbitrarily used without charge. The computed group elements are given as explicit multiplicative combinations of given group elements. Let $X_\ell = {g^{(1)}}^{r_\ell} \in \mathbb{G}_1, W_\ell = {Q_A}^{r_\ell + sh_\ell} \in \mathbb{G}_2, V_\ell = \hat{e}(X_\ell, S_{B_\ell})$ for $\ell = 1, \cdots, l$ be the group elements that $\mathcal{A}$ gets from $\mathcal{BSO}$ using the sender ID_A and recipient ID_{B_ℓ}. A computed $f_j^{(1)} \in \mathbb{G}_1$ is of the form $f_j^{(1)} = P^{a_{j,-1}^{(1)}} P_{TA}^{a_{j,0}^{(1)}} \prod_{\ell=1}^{l} {X_\ell}^{a_{j,\ell}^{(1)}}$, where the exponents $a_{j,-1}^{(1)}, \cdots, a_{j,l}^{(1)} \in \mathbb{Z}_p$ depend arbitrarily on given non-group data. A computed $f_j^{(2)} \in \mathbb{G}_2$ is of the form $f_j^{(2)} = Q_A^{a_{j,0}^{(2)}} \prod_{\ell=1}^{l} {W_\ell}^{a_{j,\ell}^{(2)}}$, where the exponents depend arbitrarily on given non-group data. A computed $f_j^{(3)} \in \mathbb{G}_T$ is of the form $f_j^{(3)} = \hat{e}(P, Q_A)^{a_{j,-1}^{(3)}} \hat{e}(P_{TA}, Q_A)^{a_{j,0}^{(3)}} \prod_{\ell=1}^{l} {V_\ell}^{a_{j,\ell}^{(3)}}$.

Powers and limitations of GGM and GGPM

Because co-CDH and one-more co-CDH are collisions in GGPM, Lemma 6 implies they are hard. The perspective is that co-CDH constitutes collisions in GGPM. The real-world interpretation of this model-based result is roughly as follows: GGM (resp. GGPM) *bans* certain operations, in the sense that it can be assumed w.l.o.g. that the generic algorithm does not use these operations. The justification is that these operations are thought to be of no help. In GGM for discrete logarithm with parameters p, g, the additions (resp. subtractions) in $\mathbb{Z}_p$ are banned. In GGM for ECDL with parameters p, base point G whose order is p, arithmetics in $\mathbb{Z}_p$ are banned. In GGPM where we have in mind the $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are all groups of elliptic curve points, the GGPM model allows point operations, arithmetics in $\mathbb{Z}_p$, but bans arithmetics in $\mathbb{Z}_p$ on the argument that they do not help.

Based on such model assumptions, GGM has been used to prove results that often cannot be proved in other models. The GGM has been used to prove the hardness of the discrete logarithm [73, 84]. It has also been used to reduce plm-uf of Schnorr or Okamoto-Schnorr blind signature to the ROS Problem [81], or the one-more discrete logarithm problem. Note that the one-more discrete logarithm problem is proven hard in the GGM by simple applications of the methods used in [81]. Based on similar model assumptions, we use GGPM to reduce plm-uf of blind signcryption to the ROS Problem or the one-more co-CDH Problem in this paper. Note that one-more co-CDH is proven hard in GGPM.

Algorithms already exist that exploit operations *banned* from GGM. The index calculus method to compute the discrete logarithm utilizes size information in $\mathbb{Z}_p$ to achieve efficiency. It is outside the boundary of GGM. In ECDL, it is suspected but not yet explicitly demonstrated that arithmetics in $\mathbb{Z}_p$ and properties of the curve can be exploited. Therefore, GGM and GGPM are used with these elliptic curves applications in mind. If and when exploitations of $\mathbb{Z}_p$ arithmetics or curve properties, or other unforeseen techniques outside the model, can be exhibited, both GGM and GGPM will need to be reexamined.

Lemma 6 also implies the hardness of the one-more co-CDH Problem in the GGPM. The *one-more co-CDH Problem* is (roughly speaking): Given q_B queries to the co-CDH Oracle, compute $q_B + 1$ co-CDH Problems.

4.5 Comparisons

The bandwidth and complexity efficiencies of our IBSC scheme is compared against a collection of existing schemes in Table 4.1.

The computation time includes the number of pairings and exponential computation as they are the most expensive in IBSC scheme. The actual number of computation which cannot be pre-computed (when the recipient identity and the message is not yet known) is shown in bracket.

For fair comparison on ciphertext size, we assume that a message m of length $|m|$ have to cut into k pieces for signcryption, usually with 160-bit for each piece. The 160-bit randomness is reused by multiple 160-bit blocks in the same message. We assume this bandwidth-conserving manoeuvre does not reduce security. We ignore the bandwidth cost of sending the sender identity by assuming it is sent just once, or not sent at all as the recipient is expecting a few senders. $|\mathbb{G}_1|$ (resp. $|\mathbb{F}_p|$) denotes the size of $\mathbb{G}_1$ (resp. F_p) element, which is about 160 bits for most representative in elliptic curve implementation and signcryption applications. In LQ2 [64], δ is 160 bits for ciphertext unlinkability, and 0 bit for ciphertext linkability.

Schemes M, LQ1, NR and CYSC are not IND-A secure, because the unsigncryption requires the knowledge of sender identity in advance.

4.5.1 Comment for IND-B

In the following, please refer to the original paper for original scheme and the definition of the symbols used. In the IND sub-game (b), the Adversary chooses message m, sender ID_A and recipient ID_{B1}. The Adversary knows the private key of ID_A. Simulator chooses a recipient ID_{B0}, and randomly picks $b \in \{0, 1\}$. Simulator signcrypts the message m from sender ID_A to recipient ID_{Bb} and returns the ciphertext to the Adversary. The Adversary

Table 4.1: Comparing bandwidth and complexity efficiencies of IBSC schemes. IND-A (resp. IND-B, IND-C) means sender anonymity (resp. recipient anonymity, message confidentiality). StE (Sign-then-Encrypt) and EtS (Encrypt-then-Sign) use ID-based encryption from [18] and ID-based signature from [33].

Scheme	Security				Ciphertext Size	Signcrypt Time		Unsigncrypt Time	
	IND			EU					
	A	B	C			#pair	#exp	#pair	#exp
EtS	×	√	√	√	$(2k+1)\|\mathbb{G}_1\|+2\|m\|$	1	4 (1)	3	1 (1)
StE	√	√	√	×	$(2k+1)\|\mathbb{G}_1\|+2\|m\|$	1	4 (1)	3	1 (1)
M [66]	×	√	×	√	$(k+1)\|\mathbb{G}_1\|+\|m\|$	1	3 (1)	4	1 (1)
LQ1 [63]	×	×	*	√	$k(\|\mathbb{G}_1\|+\|\mathbb{F}_p\|)+\|m\|$	2	2 (1)	4	1 (1)
NR [72]	×	×	*	×	$(k+1)\|\mathbb{G}_1\|+\|m\|$	1	3 (2)	3	1 (1)
B [22]	√	√	√	*	$(k+1)\|\mathbb{G}_1\|+\|m\|$	1	4 (3)	4	2 (2)
CYHC [39]	×	√	√	√	$k(\|\mathbb{G}_1\|+\|\mathbb{F}_p\|)+\|m\|$	2	2 (1)	4	1 (1)
LQ2 [64]	√	√	√	*	$(k+1)\|\mathbb{G}_1\|+\|m\|+\|\delta\|$	1	4 (3)	4	1 (1)
This scheme	√	√	√	√	$(k+1)\|\mathbb{G}_1\|+\|m\|$	1	4 (1)	3	1 (1)

has to guess b.

Libert and Quisquater's scheme 1 [63]

The Adversary has the ciphertext $\langle c, r, S\rangle$ and d_A, the private key of ID_A. The Adversary computes:

$$\begin{aligned} k_2 &= H_2(\hat{e}(S, Q_{B1})\hat{e}(d_A, Q_{B1})^r) \\ m' &= D_{k_2}(c) \end{aligned}$$

The Adversary outputs $b = 1$ if $m' = m$. Otherwise, the Adversary outputs $b = 0$. Then the Adversary wins the IND game with probability 1.

Nalla and Reddy's scheme [72]

The Adversary has the ciphertext $\langle R, S, C\rangle$ and S_A, the private key of ID_A. The Adversary computes:

$$\begin{aligned} R' &= (R||H_1(\hat{e}(Q_{B1}, S_A))||m) \\ k_A &= H''(\hat{e}(Q_{B1}, R)^{H'(R')}) \\ C' &= k_A \oplus m \end{aligned}$$

The Adversary outputs $b = 1$ if $C' = C$. Otherwise, the Adversary outputs $b = 0$. Then the Adversary wins the IND game with probability 1.

4.5.2 Comment for IND-C

[63] showed M is not IND-CCA2 secure. In the IND sub-game (c), the Adversary chooses message m_1, sender ID_A and recipient ID_B. The Adversary knows the private key of ID_A. Simulator chooses a message m_0, and randomly picks $b \in \{0, 1\}$. Simulator signcrypts the message m_b from sender ID_A to recipient ID_B and returns the ciphertext to the Adversary. The Adversary has to guess b.

Libert and Quisquater's scheme 1 [63]

The Adversary has the ciphertext $\langle c, r, S\rangle$ and d_A, the private key of ID_A. The Adversary computes:

$$\begin{aligned} P_{pub}^{x'} &= Sd_A^r \\ k_2' &= H(\hat{e}(P_pub^{x'}, Q_B) \\ c' &= E_{k_2'}(m_1) \end{aligned}$$

The Adversary outputs $b = 1$ if $c' = c$. Otherwise, the Adversary outputs $b = 0$. Then the Adversary wins the IND game with probability 1.

Nalla and Reddy's scheme [72]

The Adversary has the ciphertext $\langle R, S, C\rangle$ and S_A, the private key of ID_A. The Adversary computes:

$$\begin{aligned} R' &= (R||H_1(\hat{e}(Q_B, S_A))||m_1) \\ k_A &= H''(\hat{e}(Q_B, R)^{H'(R')}) \\ C' &= k_A \oplus m_1 \end{aligned}$$

The Adversary outputs $b = 1$ if $C' = C$. Otherwise, the Adversary outputs $b = 0$. Then the Adversary wins the IND game with probability 1.

4.5.3 Comment for EU

In the EU game, the Adversary chooses message m, sender ID_A and recipient ID_B. The Adversary knows the private key of ID_B. The Adversary returns a ciphertext σ and recipient identity ID_B to the Simulator.

Nalla and Reddy's scheme [72]

The Adversary has S_B, the private key of ID_B. The Adversary randomly chooses $a \in R$ and computes:

$$\begin{aligned} R &= {S_B}^a \\ R' &= (R||H_1(\hat{e}(S_B, Q_A))||m) \\ S &= {Q_B}^{aH'(R')} \\ k_A &= H''(\hat{e}(Q_B, S_B)^{aH'(R')}) \\ C &= k_A \oplus m \end{aligned}$$

The Adversary outputs the ciphertext $\sigma = \langle R, S, C \rangle$, sender identity ID_A and recipient identity ID_B to the Simulator.

The Simulator decrypts by computing:

$$\begin{aligned} k_B &= H''(\hat{e}(S, S_B)) \\ m &= k_B \oplus C \end{aligned}$$

The decryption succeeds. Then in verification, the Simulator computes $R' = (R||H_1(\hat{e}(S_B, Q_A))||m)$ and checks if:

$$\hat{e}(S_B, S) = \hat{e}(Q_B, R)^{H'(R')}$$

By the above construction, the ciphertext must pass the verification. Then the Adversary wins the EU game with probability 1.

Boyen's scheme has unforgeability for the signature only. It does not satisfy the unforgeability for ciphertext in our security model and also the security model of standard signcryption in [1]. LQ2 scheme is similar to Boyen's in this aspect. Our IBSC scheme avoids this controversial property of unlinkability and achieves unforgeability for ciphertext.

As we can see, our IBSC scheme is the fastest, with shortest ciphertext size and proven secure in the strongest model among the existing schemes. [2]

4.6 Additional Functionality of Our Scheme

From our new efficient IBSC scheme, we can achieve further functionalities which are useful in reality. They are the TA compatibility and forward secrecy.

4.6.1 TA Compatibility

In the reality, it is quite often that sender and recipient use different TAs. If this situation happens, our scheme can still be used without major changes.

Assume all TAs use same pairing e, hash functions and $P \in \mathbb{G}_1$. Now let Alice uses $TA1$ with master key s_1. Hence $P_{TA1} = P_{s1}$ and $S_A = {Q_A}^{s_1}$. Similarly Bob uses $TA2$ with master key s_2. Hence $P_{TA2} = P_{s2}$ and $S_B = {Q_A}^{s_2}$.

In our scheme, Sign remains unchanged. In Encrypt, $V = \hat{e}({Q_B}^r, P_{TA2})$ and others remain unchanged. Decrypt remains unchanged. In Verify, $\hat{e}(P, Y)$ $= \hat{e}({P_{TA1}}^h X, Q_A)$ and others remain unchanged. Consistency is verified as:

[2] After the completion of this research, two new pairing-based IBSC schemes are proposed in [36] and [70]. [36] has the same efficiency after pre-compute and has similar security as our IBSC scheme. [70] proposed a faster scheme with same bandwidth, but there is no security proof for it.

$$\begin{array}{llcll} V & = \hat{e}(P_{TA2}, {Q_B}^r) & \text{and} & \hat{e}(P, W) & = \hat{e}(P, {S_A}^h {Q_A}^r) \\ & = \hat{e}(P^{s2}, {Q_B}^r) & & & = \hat{e}(P, {Q_A}^{(r+hs1)}) \\ & = \hat{e}(X, S_B) & & & = \hat{e}({P_{TA1}}^h X, Q_A) \end{array}$$

The security and efficiency of our scheme remains unaffected. Therefore, our scheme can have the TA compatibility function.

4.6.2 Forward Secrecy

Our scheme can achieve forward secrecy. It means that even if the private key of the sender is compromised in the future, the past communications will not be compromised. It can be achieved as in our scheme:

$$V = \hat{e}(P_{TA}, {Q_B}^r)$$

where r cannot be known even if sender private key is compromised in the future. Therefore Adversary cannot compute V and hence cannot recover m from Z.

If sender and recipient use different TAs as in Section 6.1, then our scheme can even achieve partial TA forward secrecy. If master key of $TA1$ is compromised, then the past communications with users using different TAs will not be compromised, since the computation of V requires the knowledge of r or s_2:

$$V = \hat{e}(P_{TA2}, {Q_B}^r) = \hat{e}(P^{s_2}, {Q_B}^r)$$

Therefore even $s1$ is compromised in the future, the adversary still cannot compute V and hence cannot recover m from Z.

4.7 Chapter Conclusion

In this chapter, we have proposed a new BIBSC scheme and its security model. We introduce the generic group and pairing model (GGPM). We proof the BIBSC scheme is secure against p1m-uf in ROM+GGPM.

For the IBSC scheme, our scheme is the fastest, with shortest ciphertext and proven secure in a stronger security model when comparing with existing schemes. We provide the flexibility for choosing linkability of ciphertext or not.

Chapter 5

Identity Based Group Signatures

Group signature, introduced by Chaum and van Heyst [35], allows any member of a group to sign on behalf of the group. However, the identity of the signer is kept secret. Anyone can verify that the signature is signed by a group member, but cannot tell which one. Therefore group signature provides anonymity for signers. Usually in group signature schemes, a group manager issues certificates to his group members. Then the group member uses his certificate and his own secret key to sign messages. Anyone can verify the signature by the group manager's public key. In some cases, an open authority has a secret key to revoke the anonymity of any signature in case of dispute. Mostly it can be done by an encryption to the open authority when signing the message. On the other hand, anonymity can be revoked when a signer double signs in some schemes. Group signature is a very useful tool in real world. It can be used in e-cash, e-voting or attestation [24] in trusted computing group.

After the introduction of group signature by Chaum and van Heyst [35], there are numerous group signature schemes proposed, such as Ateniese et al. [2], Dodis et al. [44], Boneh et al. [16]. The state-of-the-art is to have a group signature scheme with signature size independent of the group size. The security model of dynamic group signature is proposed in [9].

However, none of the existing group signature scheme can be completely verified in an identity based manner, that is the group public key and the opener public key are arbitrary strings. The current "Identity based" group signature are mostly for identity based group member only ([76, 68, 87, 32, 37]). We think that identity based group member is not enough for group signature. It is because the signer's public key is always anonymous in group signature. Whether it is identity based or not has no effect to the verifier. We think that it is constructive to have a group signature with identity based group public key, which is the identity of the group manager in this case. At the same time, we also want to support identity based group members, as well as open authority. We call this new scheme to be a fully identity based group signature. However all of the existing schemes only have identity based key pairs for group members only. Group signature scheme with identity based group manager and identity based open authority remains as an open problem. In this paper, we will give a generic construction, and then a specific instantiation of such a identity based group signature.

Contributions. Our main contributions are:

- We introduce the formal study of group signature schemes with identity based group manager, identity based group members and identity based open authority.

- We present the first construction of the above scheme, complete with security models, and reductionist security proofs in the random oracle model. The size of the signature is $O(\lambda)$ bits.

- We extend Bellare, Shi, and Zhang [9]'s generic group signature construction by verifiably encrypt, to the Open Authority (OA), a one-way image of the signer public key instead of the signer public key itself. This technique is crucial to the topic in this paper.

5.1 New Intractability Assumption

Definition 33 *Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be a pairing. Given the following:*

1. $g_1, g_1^{\alpha}, g_1^{\beta_i}, g_1^{\gamma_i} \in \mathbb{G}_1$ *for* $1 \leq i \leq k$*;*

2. $g_2, g_2^{\delta_1}, g_2^{\delta_2} \in \mathbb{G}_2$*,* $R \in G_T$*;*

3. $\Pr\{\gamma_i = \alpha\beta_i, \ \textit{all}\ i, 1 \leq i \leq k\} = \Pr\{\gamma_i \neq \alpha\beta_i, \ \textit{all}\ i, 1 \leq i \leq k\} = 1/2.$

4. $\Pr\{\gamma_i = \alpha\beta_i, \ \textit{all}\ i, 1 \leq i \leq k\ \textit{AND}\ R = \hat{e}(g_1, g_2)^{\delta_1\delta_2}\} = \Pr\{\gamma_i \neq \alpha\beta_i, \ \textit{all}\ i,\ 1 \leq i \leq k\ \textit{AND}\ R \neq \hat{e}(g_1, g_2)^{\delta_1\delta_2}\} = 1/2$

The Lockstep DDH Problem *(resp.* Lockstep DDH+coDBDH Problem*) is to distinguish between the two nonzero probability events in (3) (resp. (4)) above with non-negligible probability over 1/2. The* Lockstep DDH Assumption *(resp.* Lockstep DDH+coDBDH Assumption*) is that no PPT algorithm can solve the Lockstep DDH Problem (resp. Lockstep DDH+coDBDH Problem).*

Lemma 7 *The Lockstep DDH Assumption in $\mathbb{G}_1$ holds if and only if the DDH Assumption in $\mathbb{G}_1$ holds. The Lockstep DDH+coDBDH Assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$ if and only if the DDH Assumption in $\mathbb{G}_1$ and the co-DBDH assumption in $(\mathbb{G}_2, \mathbb{G}_1)$ both hold.*

Proof. We prove for the Lockstep DDH+coDBDH Assumption only. The other case is similar. The DDH assumption and the co-DBDH assumption implies the Lockstep DDH+coDBDH assumption is straightforward.

We now proof in the opposite direction. Let $\mathcal{B}$ be a PPT solver of the Lockstep DDH+coDBDH problem with advantage ϵ_1. Consider its performance when given the following problems: [DDHi: $(g_1, g_1^{\alpha}, g_1^{\beta_i}, g_1^{\gamma_i})$] for

$1 \leq i \leq k$ and [co-DBDH: $(g_1, g_2, g_2^{\delta_1}, g_2^{\delta_2}, R)$]; where $\gamma_i = \alpha\beta_i$ or is random with half-half probability, and $R = \hat{e}(g_1, g_2)^{\delta_1\delta_2}$ or is random with half-half probability. Then we can give the "generalized lockstep" problem to $\mathcal{B}$ to solve: $[(g_1, g_1^{\alpha}, g_1^{\beta_i}, g_1^{\gamma_i})$ for $1 \leq i \leq k$; $(g_2, g_2^{\delta_1}, g_2^{\delta_2}, R)]$. With probability $2^{-(k+1)}$, the "generalized lockstep" problem is a Lockstep DDH+coDBDH problem, and in that case $\mathcal{B}$ solves it with probability $1/2 + \epsilon_1$. Otherwise, the "generalized lockstep" problem is not a Lockstep DDH+coDBDH problem, and let us consider $\mathcal{B}$'s performance in this case. Let ϵ_2 denote the probability that $\mathcal{B}$ outputs $\perp$ meaning the problem is not a Lockstep DDH+coDBDH problem. $\epsilon_2 = 0$ if he is not allowed to do so. Then $\mathcal{B}$ outputs either DDHi and co-DBDH decision with equal probability $(1 - \epsilon_2)/2$ because there is a symmetry w.r.t. the two cases.

Let us build an algorithm $\mathcal{B}$' to solve DDHi and co-DBDH: $\mathcal{B}$' outputs "yes" to DDHi and co-DBDH if $\mathcal{B}$ outputs "yes" on input "generalized lockstep" problem; and $\mathcal{B}$' outputs "no" otherwise. Then:

$$\begin{aligned} &\textstyle\sum_{i=1}^{k} \frac{1}{k+1}\Pr\{\mathcal{B}\text{' solves DDH}i\} + \frac{1}{k+1}\Pr\{\mathcal{B}'\text{ solves co-DBDH}\} \\ = &\textstyle\frac{1}{2^{k+1}}\Pr\{\mathcal{B}\text{ solves Lockstep DDH+coDBDH}\} + \epsilon_2 + \frac{1}{2}(1 - \frac{1}{2^{k+1}} - \epsilon_2) \end{aligned}$$

Therefore $\mathcal{B}$' has a probability non-negligibly over half of solving either DDHi or co-DBDH problem.

5.2 Security Model

We present a security model for the identity based group signature. Here we adapt the models for dynamic group signature in [9], and add support for IBGS. Our scheme is applicable to multiple certificate authorities (CA, or group managers) and open authorities (OA).

5.2.1 Syntax

A *identity-based group signature (IBGS)* is a tuple (Init, OKg, GKg, UKg, Join, Iss, GSig, GVf, Open, Judge) where:

- Init: $1^\lambda \mapsto \mathsf{param}$. On input the security parameter 1^λ, generates system-wide public parameters param. The identity manager of CA (IM_A) has (sk,pk) pair (x_A, y_A) for CA (resp. IM_U has (x_U, y_U) for group members, IM_O has (x_O, y_O) for OA) and an efficiently samplable one-way NP-relation $\langle \mathcal{R}_A \rangle$, with trapdoor x_A (resp. $\langle \mathcal{R}_U \rangle$, with trapdoor x_U, $\langle \mathcal{R}_O \rangle$, with trapdoor x_O). An efficiently samplable family of one-way NP-relation $\mathcal{F} = \{\langle \mathcal{R}_{C,i} \rangle : i\}$ with trapdoor gsk_i, is defined for issuing certificate. param is (y_A, y_U, y_O, $\mathcal{R}_A$, $\mathcal{R}_U$, $\mathcal{R}_O$, $\mathcal{F}$).

- OKg:$(\mathsf{oa}, x_O) \mapsto (x_{\mathsf{oa}}, aux_{\mathsf{oa}})$. On input the OA identity oa, the IM_O uses his secret key x_O to compute the secret key x_{oa} of the OA, some auxiliary information aux_{oa} such that $((x_{\mathsf{oa}}, aux_{\mathsf{oa}}), \mathsf{oa}) \in \mathcal{R}_O$.

- GKg: $(\mathsf{ca}, x_A) \mapsto (x_{\mathsf{ca}}, aux_{\mathsf{ca}}, \langle \mathcal{R}_{C,\mathsf{ca}} \rangle)$. On input the CA identity ca, the IM_A samples $\mathcal{F}$ to get the relation $\langle \mathcal{R}_{C,\mathsf{ca}} \rangle$. The IM_A uses his secret key x_A to compute the group secret key of CA x_{ca}, some auxiliary information aux_{ca} such that $((x_{\mathsf{ca}}, aux_{\mathsf{ca}}), \mathsf{ca}) \in \mathcal{R}_A$.

- UKg: $(\mathsf{id}, x_U) \mapsto (x_{\mathsf{id}}, aux_{\mathsf{id}})$. On input the identity id of the member, the IM_U uses his secret key x_U to compute the secret key x_{id} of the member, some auxiliary information aux_{id} such that $((x_{\mathsf{id}}, aux_{\mathsf{id}}), \mathsf{id}) \in \mathcal{R}_U$.

- Join,Iss is a pair of interactive protocols between the user and the CA, with common inputs ca and id. Iss's additional inputs are x_{ca} and aux_{ca}. Join's additional inputs are x_{id} and aux_{id}. At the conclusion, Join obtains $cert_{\mathsf{id}}$ satisfying $((x_{\mathsf{id}}, aux_{\mathsf{id}}, cert_{\mathsf{id,ca}}), \mathsf{id}) \in \mathcal{R}_{C,\mathsf{ca}}$, and Iss stores $(\mathsf{id}, cert_{\mathsf{id,ca}})$ in a registration table reg.

- GSig: (id, x_{id}, aux_{id}, ca, oa, $cert_{\mathsf{id,ca}}$, M) $\mapsto \sigma$. On input the keys, certificates and message, outputs a signature σ.

- GVf: $(\mathsf{ca}, \mathsf{oa}, M, \sigma) \mapsto 0$ or 1. On input the message and signature, outputs 1 for valid signature and 0 for invalid signature.

- Open: $(\mathsf{ca}, x_{\mathsf{oa}}, \mathsf{reg}, M, \sigma) \mapsto (i, \omega)$. The OA with key x_{oa} has read access to reg. On input a valid signature σ for message M for ca, output identity i for the corresponding signer, and ω is the proof of this claim. Output $i = \perp$ if no such member is found.

- Judge: $(\mathsf{ca}, \mathsf{id}, \mathsf{oa}, M, \sigma, \omega) \mapsto 0$ or 1. It checks if the proof ω is a valid proof that id is the real signer of σ for message M under ca, oa. Outputs 1 for valid and 0 for invalid.

Remarks: Here we use (param, ca) to denote gpk in [9]'s original syntax. We also split the GKg in [9] into Init, OKg and GKg. It is because we want to emphasize that group managers (CA) and open authorities (OA) are identity based.

5.2.2 Security Notions

We have the security notions of Correctness, Anonymity, Traceability, Non-frameability from [9], with modification for identity based. We give a brief description here.

Correctness: Let $\sigma \leftarrow$ GSig(id, x_{id}, aux_{id}, ca, oa, $cert_{\mathsf{id,ca}}$, M) for arbitrary id, x_{id}, aux_{id}, ca, oa, $cert_{\mathsf{id,ca}}$, M. The IBGS has opening correctness if $(\mathsf{id}, \omega) \leftarrow \mathsf{Open}(\mathsf{ca}, x_{\mathsf{oa}}, \mathsf{reg}, M, \sigma)$ and $\mathsf{Judge}(\mathsf{ca}, \mathsf{id}, \mathsf{oa}, M, \sigma, \omega) = 1$ with overwhelming probability. It has verification correctness if $\mathsf{GVf}(\mathsf{ca}, \mathsf{oa}, M, \sigma) = 1$

with probability 1. The IBGS is correct if it has verification and opening correctness.

We have the following oracles for the adversary to query:

- The *Random Oracle* $\mathcal{RO}$: simulate the random oracle normally.

- The *Key Extraction Oracle-CA* $\mathcal{KEO}_c$: $\mathsf{ca} \rightarrow x_{\mathsf{ca}}$. Upon input CA ca, outputs his secret key x_{ca}.

- The *Key Extraction Oracle-OA* $\mathcal{KEO}_o$: $\mathsf{oa} \rightarrow x_{\mathsf{oa}}$. Upon input OA oa, outputs his secret key x_{oa}.

- The *Key Extraction Oracle-User* $\mathcal{KEO}_u$: $\mathsf{id} \rightarrow x_{\mathsf{id}}$. Upon input user id, outputs his secret key x_{id}.

- The *Join Oracle* $\mathcal{JO}$: $(\mathsf{id}, \mathsf{ca}) \rightarrow cert_{\mathsf{ca}}$. Upon input id of group ca, outputs the $cert_{\mathsf{ca}}$ corresponding to an honest Iss-executing CA.

- The *Issue Oracle* $\mathcal{IO}$: $(\mathsf{id}, \mathsf{ca}) \rightarrow cert_{\mathsf{ca}}$. Upon input id of group ca, outputs the $cert_{\mathsf{ca}}$ corresponding to an honest Join-executing user.

- The *Corruption Oracle* $\mathcal{CO}$: $(\mathsf{id}, \mathsf{ca}) \rightarrow (x_{\mathsf{id}}, aux_{\mathsf{id}}, cert_{\mathsf{ca}})$. Upon input user id of group ca, outputs the secret keys $(x_{\mathsf{id}}, aux_{\mathsf{id}}, cert_{\mathsf{ca}})$.

- The *Signing Oracle* $\mathcal{SO}$: $(\mathsf{id}, \mathsf{ca}, \mathsf{oa}, M) \rightarrow \sigma$. Upon input user id, group ca, oa and a message M, outputs a valid signature.

- The *Open Oracle* $\mathcal{OO}$: $(\mathsf{oa}, \mathsf{ca}, M, \sigma) \rightarrow (\mathsf{id}, \omega)$. Upon input a valid signature σ for message M under ca, oa, outputs the signer id and the proof ω.

  Remark: $\mathcal{KEO}_O$ is a stronger oracle than $\mathcal{OO}$ in the sense that $\mathcal{KEO}_O$ directly gives the secret key for OA, while $\mathcal{OO}$ only opens a particular signature.

Anonymity: We have the following **Experiment Anon** for anonymity:

1. Simulator $\mathcal{S}$ invokes Init. $\mathcal{S}$ invokes UKg, Join, Iss together q_u times to generate a set of honest users, denoted HU, with secret keys and certificates.

2. $\mathcal{A}$ queries $\mathcal{RO}, \mathcal{CO}, \mathcal{OO}, \mathcal{IO}, \mathcal{KEO}_c, \mathcal{KEO}_u, \mathcal{KEO}_o$ in arbitrary interleaf.

3. $\mathcal{A}$ selects two users $\mathsf{id}_0, \mathsf{id}_1 \in$ HU, $\mathsf{ca}_g, \mathsf{oa}_g$ a message M and gives them to $\mathcal{S}$. Then $\mathcal{S}$ randomly chooses $b \in \{0,1\}$ and returns the gauntlet ciphertext $\sigma \leftarrow \mathcal{SO}(\mathsf{id}_b, \mathsf{ca}_g, \mathsf{oa}_g, M)$. oa_g should not be input to $\mathcal{OO}, \mathcal{KEO}_o$ before.

4. $\mathcal{A}$ queries $\mathcal{RO}, \mathcal{CO}, \mathcal{OO}, \mathcal{IO}, \mathcal{KEO}_c, \mathcal{KEO}_u, \mathcal{KEO}_o$ in arbitrary interleaf. oa_g should not be input to $\mathcal{OO}, \mathcal{KEO}_o$.

5. $\mathcal{A}$ delivers an estimate $\hat{b} \in \{0,1\}$ of b.

$\mathcal{A}$ also has write access to registration table reg in the experiment. $\mathcal{A}$ wins the Experiment Anon if $\hat{b} = b$, and oa_g has never been queried to $\mathcal{KEO}_o$. $\mathcal{A}$'s advantage is its probability of winning Experiment Anon minus half.

Remark: By not allowing to query the gauntlet oa_g, our model is closer to that of [16] which does not support any $\mathcal{OO}$, than to that of [9] which supports $\mathcal{OO}$.

Definition 34 *The IBGS is anonymous if no PPT adversary has a non-negligible advantage in Experiment Anon.*

Traceability: We have the following **Experiment Trace** for traceability:

1. $\mathcal{S}$ invokes Init. $\mathcal{S}$ invokes UKg, Join, Iss together q_u times to generate a set of honest users, denoted HU, with secret keys and certificates.

2. $\mathcal{A}$ queries $\mathcal{RO}, \mathcal{CO}, \mathcal{JO}, \mathcal{KEO}_c, \mathcal{KEO}_u, \mathcal{KEO}_o$ in arbitrary interleaf.

3. $\mathcal{A}$ delivers signature σ for messages M for group ca and open authority oa. ca should not be input to $\mathcal{KEO}_c$.

$\mathcal{A}$ also has read access to reg. $\mathcal{A}$ wins the Experiment Trace if $\mathsf{GVf}(\mathsf{ca}, \mathsf{oa}, M, \sigma) = 1$, either $i = \perp$ or $\mathsf{Judge}(\mathsf{ca}, i, \mathsf{oa}, m, \sigma, \omega) = 0$, where $(i, \omega) \leftarrow \mathsf{Open}(\mathsf{ca}, x_{\mathsf{oa}}, \mathsf{reg}, M, \sigma)$, ca has never been queried to $\mathcal{KEO}_c$, and (i, ca) has never been queried to $\mathcal{CO}$, $\mathcal{A}$'s advantage is its probability of winning.

Definition 35 *The IBGS is traceable if no PPT adversary has a non-negligible advantage in Experiment Trace.*

Non-Frameability: We have the following **Experiment NF** for non-frameability:

1. $\mathcal{S}$ invokes Init. $\mathcal{S}$ invokes UKg, Join, Iss together q_u times to generate a set of honest users, denoted HU, with secret keys and certificates.

2. $\mathcal{A}$ queries $\mathcal{RO}, \mathcal{CO}, \mathcal{SO}, \mathcal{IO}, \mathcal{KEO}_c, \mathcal{KEO}_u, \mathcal{KEO}_o$ in arbitrary interleaf.

3. $\mathcal{A}$ delivers (σ, M, i, ω), where ω is the proof of user i signed the signature σ for messages M with group ca and open authority oa.

$\mathcal{A}$ also has write access to reg. $\mathcal{A}$ wins the Experiment NF if $\mathsf{GVf}(\mathsf{ca}, \mathsf{oa}, M, \sigma) = 1$, $\mathsf{Judge}(\mathsf{ca}, i, \mathsf{oa}, M, \sigma, \omega) = 1$, i has never been queried to $\mathcal{CO}$ and σ is not the output from $\mathcal{SO}$ for $M, i, \mathsf{ca}, \mathsf{oa}$. $\mathcal{A}$'s advantage is its probability of winning.

Definition 36 *The IBGS is non-frameable if no PPT adversary has a non-negligible advantage in Experiment NF.*

Definition 37 *An IBGS scheme is secure if it is correct, anonymous, traceable and non-frameable.*

5.3 Constructions

In this chapter, we present a generic construction for *identity-based group signature (IBGS)* which is applicable to different kinds of relations between the identity based CA, users and open authority. After the generic construction, we give an efficient implementation which is provably secure in the random oracle model.

5.3.1 Generic Construction

A generic *IBGS* is a tuple (Init, OKg, GKg, UKg, Join, Iss, GSig, GVf, Open, Judge):

- Init, GKg, OKg, UKg, Open, Judge follows the syntax.
- Join,Iss is a pair of interactive protocols with common inputs ca and id. Iss's additional inputs are x_{ca} and aux_{ca}. Join's additional inputs are x_{id} and aux_{id}. Join runs a proof of knowledge protocol to proof that he knows x_{id} and aux_{id} to Iss. At the conclusion, Join obtains $cert_{\mathsf{id}}$ satisfying $((x_{\mathsf{id}}, aux_{\mathsf{id}}, cert_{\mathsf{id,ca}}), \mathsf{id}) \in \mathcal{R}_{C,\mathsf{ca}}$, and Iss stores $(\mathsf{id}, cert_{\mathsf{id,ca}})$ in a registration table reg. Join may also obtain aux_{ca} as part of $cert_{\mathsf{id,ca}}$.
- GSig: (id, ca, oa, x_{id}, aux_{id}, $cert_{\mathsf{id}}$, M) $\mapsto \sigma$. A user id who has $cert_{\mathsf{id}}$ runs:

$$SPK\{(\mathsf{id}, x_{\mathsf{id}}, aux_{\mathsf{id}}, cert_{\mathsf{id,ca}}, r) : (x_{\mathsf{id}}, aux_{\mathsf{id}}, \mathsf{id}) \in \mathcal{R}_U$$
$$\wedge \ (\mathsf{id}, aux_{\mathsf{id}}, cert_{\mathsf{id,ca}}) \in \mathcal{R}_{C,\mathsf{ca}} \ \wedge \ \mathsf{ctxt} = Enc(\mathsf{id}, \mathsf{oa}, r)\}(M)$$

  The signature σ is obtained from the above "signature from proof-of-knowledge" SPK, following [28]'s notion.

- GVf: $(\sigma, M) \mapsto 0$ or 1. On input the signature σ, a verifier verifies σ according to the above SPK. The verifier outputs 1 for valid signature and 0 otherwise.

5.3.2 An Instantiation: IBGS-SDH

We instantiate the generic construction above in the SDH group.

Init: On input the security parameter 1^λ, generates a pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ where the above three (multiplicative) cyclic groups are of order p. The IM_A (resp. IM_O, IM_U) secret key is $x_A \in \mathbb{Z}_p^*$ (resp. x_O, x_U) and public keys are $g_A, y_A = g_A^{x_A} \in \mathbb{G}_2$ (resp. $g_O, y_O = g_O^{x_O}$, and $g_U, y_U = g_U^{x_U}$). Let u be a generator in $\mathbb{G}_1$. Define cryptographic hash functions $\mathcal{H}_A : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, $\mathcal{H}_U : \{0,1\}^* \rightarrow \mathbb{G}_1$, $\mathcal{H}_O : \{0,1\}^* \rightarrow \mathbb{G}_1$, $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$.

For CA, define $\mathcal{R}_A = \{((x_{\mathsf{ca}}, R), \mathsf{ca}) : g_A^{x_{\mathsf{ca}}} = R y_A^{\mathcal{H}_A(R||\mathsf{ca})}\}$. For OA, define $\mathcal{R}_O = \{(x_{\mathsf{oa}}, \mathsf{oa}) : x_{\mathsf{oa}} = \mathcal{H}_O(\mathsf{oa})^{x_O}\}$. For user, define $\mathcal{R}_U = \{(x, i) : x = \mathcal{H}_U(i)^{x_U}\}$. For certificate, define $\mathcal{F} = \{\langle \mathcal{R}_{C,i} \rangle : i\}$ with trapdoor x_i. $\mathcal{R}_{C,\mathsf{ca}} = \{(\mathsf{id}, (A, e)) : A^{e + x_{\mathsf{ca}}} \mathcal{H}_U(\mathsf{id}) = u\}$.

Let $g_0, g_1, g_2, g_3, g_4, u$ are generators in $\mathbb{G}_1$. Then:
$\mathsf{param} = (\hat{e}, g_A, y_A, g_O, y_O, g_U, y_U, g_0, \ldots, g_4, u, \mathcal{H}_A, \mathcal{H}_U, \mathcal{H}_O, \mathcal{H}, \mathcal{R}_A, \mathcal{R}_U, \mathcal{R}_O, \mathcal{F})$.

OKg: On input OA identity oa, the identity manager IM_O uses x_O to compute OA secret key $x_{\mathsf{oa}} = \mathcal{H}_O(\mathsf{oa})^{x_O}$.

GKg: On input CA identity ca, the identity manager IM_A defines $\mathcal{R}_{C,\mathsf{ca}} = \{(\mathsf{id}, (A, e)) : A^{e + x_{\mathsf{ca}}} \mathcal{H}_U(\mathsf{id}) = u\}$ and computes as follows:

1. Randomly generate $r \in \mathbb{Z}_p^*$.

2. Compute $aux_{\mathsf{ca}} = g_A^r$, $x_{\mathsf{ca}} = r + \mathcal{H}_A(aux_{\mathsf{ca}}||\mathsf{ca}) x_A \bmod p$.

This is taken from BNN-IBI [6]. Finally CA gets $(x_{\mathsf{ca}}, aux_{\mathsf{ca}})$.

UKg: On input user identity id, the identity manager IM_U uses x_U to compute user secret key $x_{\mathsf{id}} = \mathcal{H}_U(\mathsf{id})^{x_U}$.

$\mathsf{Join}, \mathsf{Iss}$: Common inputs are id, ca. Join's additional input is x_{id} and Iss's additional inputs are $x_{\mathsf{ca}}, aux_{\mathsf{ca}}$. Join firstly runs a proof of knowledge of x_{id} for id. Then Iss uses x_{ca}, aux_{ca} to computes $cert_{\mathsf{id,ca}} = (A, e)$ satisfying $(\mathsf{id}, cert_{\mathsf{id,ca}}) \in \mathcal{R}_{C,\mathsf{ca}}$. Iss randomly selects $e \in \mathbb{Z}_p^*$, and computes $A = (u/\mathcal{H}_U(\mathsf{id}))^{1/(e+x_{\mathsf{ca}})}$. Iss sends $(A, e, aux_{\mathsf{ca}})$ to Join. The validity of aux_{ca} can be checked by BNN's IBI [6]. Note u is a fairly generated public parameter, Join accepts the certificate if and only if

$$\hat{e}(u, g_A) = \hat{e}(A, g_A)^e \hat{e}(A, S) \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A),$$

where $S = g_A^{x_{\mathsf{ca}}} = aux_{\mathsf{ca}} y_A^{\mathcal{H}_A(aux_{\mathsf{ca}} \| \mathsf{ca})}$. Finally Join obtains $cert_{\mathsf{id,ca}}, aux_{\mathsf{ca}}$. Iss computes $W = \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A)$, and puts (id, A, e, W) in reg.

GSig: A member id from group ca with secret key x and certificate (A, e) computes a signature σ for message M and oa by

$$\begin{aligned} SPK\{(\mathsf{id}, x, (A, e), d) : x = \mathcal{H}_U(\mathsf{id})^{x_U} \ \wedge \ A^{e+x_{\mathsf{ca}}} \mathcal{H}_U(\mathsf{id}) = u \\ \wedge \ \mathsf{ctxt} = \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A) \hat{e}(\mathcal{H}_O(\mathsf{oa}), y_O)^d \ \wedge \ U = g_O^d\}(M) \end{aligned}$$

which is equivalent to

$$\begin{aligned} SPK\{(\mathsf{id}, x, (A, e), d) &: \hat{e}(x, g_U) = \hat{e}(\mathcal{H}_U(\mathsf{id}), y_U) \\ &\wedge \ \hat{e}(u, g_A) = \hat{e}(A, g_A)^e \hat{e}(A, S) \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A) \\ &\wedge \ \mathsf{ctxt} = \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A) \hat{e}(\mathcal{H}_O(\mathsf{oa}), y_O)^d \ \wedge \ U = g_O^d \\ &\wedge \ S = aux_{\mathsf{ca}} y_A^{\mathcal{H}_A(aux_{\mathsf{ca}} \| \mathsf{ca})}\}(M) \end{aligned} \tag{5.1}$$

The further instantiation is as follows. Randomly selects $s_1, d \in \mathbb{Z}_p^*$. Computes $s_2 = es_1$. The masked images are:

$$t_0 = g_0^{s_1} \ \wedge\ t_1 = xg_1^{s_1} \ \wedge\ t_2 = \mathcal{H}_U(\mathsf{id})g_2^{s_1} \ \wedge\ t_3 = Ag_3^{s_1} \ \wedge\ t_5 = t_3^e g_4^{s_1}$$

And we have: $\mathsf{ctxt} = \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A)\hat{e}(\mathcal{H}_O(\mathsf{oa}), y_O)^d \ \wedge\ U = g_O^d$.

Randomly selects $r_1, r_2, r_3, r_4 \in \mathbb{Z}_p$, $R_1, R_2, R_3 \in \mathbb{G}_1$. Computes:

$$\begin{aligned} & \tau_0 = g_0^{r_1} \ \wedge\ \tau_1 = R_1 g_1^{r_1} \ \wedge\ \tau_2 = R_2 g_2^{r_1} \ \wedge\ \tau_3 = R_3 g_3^{r_1} \\ \wedge\ \ & \tau_4 = [\hat{e}(g_1, g_U)^{-1}\hat{e}(g_2, y_U)]^{r_1} \ \wedge\ \tau_5 = t_3^{r_3} g_4^{r_1} \\ \wedge\ \ & \tau_6 = \hat{e}(g_3, g_A)^{r_2}[\hat{e}(g_3, S)\hat{e}(g_2 g_4, g_A)]^{r_1} \ \wedge\ \tau_7 = g_A^{r_4} \\ \wedge\ \ & \tau_8 = \hat{e}(\mathcal{H}_O(\mathsf{oa}), y_O)^{r_4}\hat{e}(g_2, g_A)^{-r_1} \end{aligned}$$

The challenge is:

$$c = \mathcal{H}((t_0, \cdots, t_3, t_5)||(\tau_0, \cdots, \tau_8)||aux_{\mathsf{ca}}||\mathsf{ctxt}||U||M) \tag{5.2}$$

The responses are:

$$\begin{aligned} & z_0 = r_1 - cs_1 \ \wedge\ Z_1 = R_1 x^{-c} \ \wedge\ Z_2 = R_2 \mathcal{H}_U(i)^{-c} \\ \wedge\ \ & Z_3 = R_3 A^{-c} \ \wedge\ z_4 = r_3 - ce \ \wedge\ z_5 = r_2 - cs_2 \ \wedge\ z_6 = r_4 - cd \end{aligned}$$

The signature σ is: $(t_0, \cdots, t_3, t_5)||c||(z_0, \cdots, z_6)||aux_{\mathsf{ca}}||\mathsf{ctxt}||U||M$.

GVf: Given a signature σ, it computes:

$$\begin{aligned} & t_4 = \hat{e}(t_1, g_U)^{-1}\hat{e}(t_2, y_U) \ \wedge\ t_6 = \hat{e}(u, g_A)^{-1}\hat{e}(t_2 t_5, g_A)\hat{e}(t_3, S) \\ \wedge\ \ & t_8 = \mathsf{ctxt} \cdot \hat{e}(t_2, g_A)^{-1} \ \wedge\ \tau_0 = g_0^{z_0} t_0^c \ \wedge\ \tau_1 = Z_1 g_1^{z_0} t_1^c \\ \wedge\ \ & \tau_2 = Z_2 g_2^{z_0} t_2^c \ \wedge\ \tau_3 = Z_3 g_3^{z_0} t_3^c \ \wedge\ \tau_4 = [\hat{e}(g_1, g_U)^{-1}\hat{e}(g_2, y_U)]^{z_0} t_4^c \\ \wedge\ \ & \tau_5 = t_3^{z_4} g_4^{z_0} t_5^c \ \wedge\ \tau_6 = \hat{e}(g_3, g_A)^{z_5}[\hat{e}(g_3, S)\hat{e}(g_2 g_4, g_A)]^{z_0} t_6^c \ \wedge\ \tau_7 = g_A^{z_6} U^c \\ \wedge\ \ & \tau_8 = \hat{e}(\mathcal{H}_O(\mathsf{oa}), y_O)^{z_6}\hat{e}(g_2, g_A)^{-z_0} t_8^c \ \wedge\ S = aux_{\mathsf{ca}} y_A^{\mathcal{H}_A(aux_{\mathsf{ca}}||\mathsf{ca})} \end{aligned}$$

Then it computes challenge $\hat{c}$ according to Eq. (5.2), and compares it to the challenge c received in the signature. If they are equal, output 1 for *valid signature*. In all other cases, output 0.

Open: The open authority uses his secret key x_{oa} to open the encryption in the signature σ. Denote $Q_{\mathsf{oa}} = \mathcal{H}_O(\mathsf{oa})$. He computes:

$$m = \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A) = \mathsf{ctxt}/\hat{e}(x_{\mathsf{oa}}, U)$$

The open authority compares W with the registration table reg. If no such entry is find, output $\perp$. If it is found to be user id, the open authority computes a proof of knowledge of x_{oa} such that $\hat{e}(x_{\mathsf{oa}}, U) = \mathsf{ctxt}/m$:

1. Randomly picks $s'_0 \in \mathbb{Z}_p$. Computes:

$$t'_0 = x_{\mathsf{oa}} h^{s'_0} \ \wedge \ t'_1 = \hat{e}(h, U)^{s'_0} \ \wedge \ t'_2 = \hat{e}(h, g_O)^{s'_0}.$$

2. Randomly picks $r'_0, r'_1 \in \mathbb{Z}_p$. Computes:

$$\tau'_0 = Q_{\mathsf{oa}}^{r'_1} h^{r'_0} \ \wedge \ \tau'_1 = \hat{e}(h, U)^{r'_0} \ \wedge \ \tau'_2 = \hat{e}(h, g_O)^{r'_0}.$$

3. Computes $c' = \mathcal{H}((t'_0, t'_1, t'_2)||(\tau'_0, \tau'_1, \tau'_2)||\mathsf{ctxt}||U||m)$.

4. Computes $z'_0 = r'_0 - c's'_0, Z'_1 = Q_{\mathsf{oa}}^{r'_1} x_{\mathsf{oa}}^{c'}$.

Outputs the proof $\omega = (t'_0||c'||(z'_0, Z'_1))$ to judge.

Judge: On input $\mathsf{id}, \mathsf{ca}, \mathsf{oa}$, the signature σ and the proof ω, it computes:

$$\begin{aligned} & m = \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A) \ \wedge \ m' = \mathsf{ctxt}/m \\ \wedge \ & t'_1 = \hat{e}(t'_0, U)/m' \ \wedge \ t'_2 = \hat{e}(t'_0, g_O)\hat{e}(Q_{\mathsf{oa}}, y_O) \\ \wedge \ & \tau'_0 = Z'_1 {t'_0}^{c'} h^{z'_0} \ \wedge \ \tau'_1 = \hat{e}(h, U)^{z'_0} {t'_1}^{c'} \ \wedge \ \tau'_2 = \hat{e}(h, g_O)^{z'_0} {t'_2}^{c'} \end{aligned}$$

Then compares if $c' = \mathcal{H}((t'_0, t'_1, t'_2)||(\tau'_0, \tau'_1, \tau'_2)||\mathsf{ctxt}||U||m)$. If it is true, output 1. Otherwise, output 0.

5.4 Security Theorems

We now give the security theorems for the above instantiation. It follows the definition in section 5.2. The proofs can be found in the full version of this paper.

Theorem 6 *The IBGS-SDH scheme is correct.*

Proof. Obvious.

Theorem 7 *The IBGS-SDH is anonymous in the random oracle model if and only if the DDH Assumption in $\mathbb{G}_1$ and the co-DBDH Assumption in $(\mathbb{G}_2, \mathbb{G}_1)$ both hold.*

Proof. Suppose $\mathcal{A}$ is a PPT algorithm that breaks the anonymity of the group signature. Then we show how to construct a PPT algorithm $\mathcal{S}$ that solves the Lockstep DDH+coDBDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$, which is equivalent to the co-DBDH problem in $(\mathbb{G}_2, \mathbb{G}_1)$ and the DDH problem in $\mathbb{G}_1$.

$\mathcal{S}$ is given the instance $g'_1, {g'_1}^{\alpha}, {g'_1}^{\beta_i}, {g'_1}^{\gamma_i} \in \mathbb{G}_1$ for $1 \le i \le 4$; $g'_2, {g'_2}^{\delta_1}, {g'_2}^{\delta_2} \in \mathbb{G}_2$ and $R \in \mathbb{G}_T$ for unknown $\alpha_i, \beta_i, \delta_1, \delta_2 \in \mathbb{Z}_p$. $\mathcal{S}$ is sets the public parameter $g_O = g'_2, y_O = {g'_2}^{\delta_1}, g_0 = g'_1, g_1 = {g'_1}^{\beta_1}, g_1 = 2 = {g'_1}^{\beta_2}, g_3 = {g'_1}^{\beta_3}, g_4 = {g'_1}^{\beta_4}$. $\mathcal{S}$ generates $g_A, x_A, y_A = g_A^{x_A}$, $g_U, x_U, y_U = g_U^{x_U}$ and $u = g_A$. $\mathcal{S}$ randomly picks $\ell \in \{1, ..., q_H\}$, where q_H is the number of query to $\mathcal{H}_O$. $\mathcal{S}$ provides $\mathcal{A}$ the parameters param.

The oracles are simulated as follows:

- $\mathcal{H}$ is random oracle.

- $\mathcal{H}_A(aux_i||i)$: On input new aux_i, i, randomly pick $\lambda \in \mathbb{Z}_p$ Return λ. Store (aux_i, i, λ) in tape $\mathcal{L}_A$.

- $\mathcal{H}_U(i)$: On input new i, randomly pick $\lambda \in \mathbb{Z}_p$ and return g_U^λ. Store (i, λ) in tape $\mathcal{L}_U$.

- $\mathcal{H}_O(i)$: On input new i, randomly pick $\lambda \in \mathbb{Z}_p$ and return g_O^λ. Store (i, λ) in tape $\mathcal{L}_O$. For the ℓ-th query, return $Q = g_1'$ and back patch (i, Q) in $\mathcal{L}_O$. Denote this identity as i_g.

- $\mathcal{KEO}_u(i)$: Computes $\mathcal{H}_U(i)$. Then $x_i = y_U^\lambda$, where $(i, \lambda) \in \mathcal{L}_U$.

- $\mathcal{KEO}_c(ca)$: On input ca, randomly pick $h, x_{ca} \in \mathbb{Z}_p$ and computes $aux_{ca} = g_A^{x_{ca}} y_A^{-h}$. $\mathcal{S}$ back patches $\mathcal{H}_A(aux_{ca}||ca) = h$. Store $(aux_{ca}$, ca, $h)$ in tape $\mathcal{L}_A$. Return (x_{ca}, aux_{ca}).

- $\mathcal{KEO}_o(oa)$: Computes $\mathcal{H}_O(oa)$. Then $x_{oa} = y_O^\lambda$, where $(oa, \lambda) \in \mathcal{L}_O$. If $oa = i_g$, declare failure and exit.

- $\mathcal{IO}(i, ca)$: It interacts with the honest user i. Computes (x_{ca}, aux_{ca}) as in $\mathcal{KEO}_c(ca)$. Randomly selects $e \in \mathbb{Z}_p$, and computes:

$$A = (u/\mathcal{H}_U(i))^{1/(e+x_{ca})}, \quad W = \hat{e}(\mathcal{H}_U(i), g_A).$$

  Stores (i, A, e, W) in reg. Returns (A, e, aux_{ca}) to honest user i.

- $\mathcal{CO}(i, ca)$: On input the identity, this oracle outputs the user's secret keys. Computes $\mathcal{H}_1(i)$. Computes x_i as in $\mathcal{KEO}_u(i)$. Computes $cert_{i,ca}$ as in $\mathcal{IO}(i, ca)$. Returns $(x_i, cert_{i,ca})$.

- $\mathcal{OO}(\mathsf{oa}, \mathsf{ca}, m, \sigma)$: Computes $\mathcal{H}_1(\mathsf{oa})$. Then $x_{\mathsf{oa}} = y_A^\lambda$, where $(\mathsf{oa}, \lambda) \in \mathcal{L}_1$. Return $(i, \omega) \leftarrow \mathsf{Open}(\mathsf{ca}, x_{\mathsf{oa}}, \mathsf{reg}, m, \sigma)$. If $oa = i_g$, declare failure and exit.

At any time, $\mathcal{A}$ can query the oracles above. At some point, $\mathcal{A}$ sends the gauntlet identity i_0, i_1, group ca, open authority oa and message M to

$\mathcal{S}$. $\mathcal{S}$ flips a coin $b \in \{0,1\}$ and computes $(x_b, A_b, e_b) \leftarrow \mathcal{CO}(i_b, ca)$. $\mathcal{S}$ sets $t_0 = g_1'^{\alpha}, t_1 = x_b g_1'^{\gamma_1}, t_2 = \mathcal{H}_U(i_b) g_1'^{\gamma_2}, t_3 = A_b g_1'^{\gamma_3}, t_5 = t_3^{e_b} g_1'^{\gamma_4}$. $\mathcal{S}$ randomly chooses a challenge $c \in \mathbb{Z}_p$ and response $z_0, ..., z_6$ from suitable domains. It computes $\tau_0, ..., \tau_8$ as in GVf. $\mathcal{S}$ sets $U = g_2'^{\delta_2}$ and computes $\mathsf{ctxt} = \hat{e}(\mathcal{H}_U(i_b), g_A)R$. Then back patch c to $\mathcal{H}$ as Eq. 5.2. $\mathcal{S}$ returns the signature σ_g as the gauntlet to $\mathcal{A}$.

$\mathcal{A}$ finally outputs a bit $\hat{b}$. If $\hat{b} = b$, $\mathcal{S}$ returns "yes" for the Lockstep DDH+coDBDH problem. Otherwise, $\mathcal{S}$ returns "no". By the back patch above, if $\mathcal{A}$ has a non-negligible advantage ε in winning the game, $\mathcal{S}$ has advantage ε / q_H in solving the Lockstep DDH+coDBDH problem, and hence can either solve the DDH problem in $\mathbb{G}_1$ or the co-DBDH problem in $(\mathbb{G}_2, \mathbb{G}_1)$.

Bow we derive the opposite reduction in the Theorem statement: Give the Adversary a Lockstep DDH+coDBDH Oracle which can solve the Lockstep DDH+coDBDH Problem, and then show it can crack Anonymity. If the adversary is given a signature

$$\sigma = (t_0, \cdots, t_3, t_5)||c||(z_0, \cdots, z_6)||aux_{\mathsf{ca}}||\mathsf{ctxt}||U||M$$

which can pass GVf. $\mathcal{A}$ is also given (x_b, A_b, e_b) for users id_b where $b \in \{0,1\}$. Then $\mathcal{A}$ randomly flips a coin $b = 0/1$ and inputs to the Oracle: $g_1' = g_0$, $g_1'^{\alpha} = t_0$, $g_1'^{\beta_1} = g_1$, $g_1'^{\gamma_1} = t_1/x_b$, $g_1'^{\beta_2} = g_2$, $g_1'^{\gamma_2} = t_2/\mathcal{H}_U(id_b)$, $g_1'^{\beta_3} = g_3$, $g_1'^{\gamma_3} = t_3/A_b$, $g_1'^{\beta_4} = g_4$, $g_1'^{\gamma_4} = t_5/t_3^{e_b}$, $g_2' = g_O$, $g_2'^{\delta_1} = y_O$, $g_2'^{\delta_2} = U$, $R = \mathsf{ctxt}/\hat{e}(\mathcal{H}_U(id_b), g_A)$. If the Oracle outputs 1, then $\mathcal{A}$ outputs id_b as the signer. Otherwise, $\mathcal{A}$ outputs id_{1-b} as the signer.

Theorem 8 *The IBGS-SDH is traceable in the random oracle model if and only if the k-CAA2 assumption holds.*

Proof. Let $\mathcal{A}$ be a PPT adversary attacking the traceability. We show that given a colluding group of k signers, with the knowledge of the opening key and access to some oracles, we can use $\mathcal{A}$ to solve the k-CAA2 problem.

$\mathcal{S}$ is given the tuple $u, v \in \mathbb{G}_1$, $g_2, g_2{}^{\gamma} \in \mathbb{G}_2$ and pairs (A_i, e_i, λ_i) with distinct and nonzero e_i's satisfying $A_i^{\gamma+e_i} v^{\lambda_i} = u$ for $1 \leq i \leq k$ as input. The value $s = log_u(v)$ is also given to $\mathcal{S}$.

$\mathcal{S}$ sets $g_A = g_2, g_U = v$. $\mathcal{S}$ randomly selects $x_A, y_A = g_A^{x_A}$, $x_U, y_U = g_U^{x_U}$ and $g_O, x_O, y_O = g_O^{x_O}$. $\mathcal{S}$ randomly selects μ and sets $g_3 = v^{\mu}$. $\mathcal{S}$ setups the rest of param and provides to $\mathcal{A}$. $\mathcal{S}$ randomly picks $\ell \in \{1, ..., q_c\}$, where q_c is number of query to $\mathcal{CO}$.

The oracles are simulated as follows:

- $\mathcal{H}_U(i)$: On input new i, randomly pick λ_j from the given k-CAA2 tuple and return v^{λ_j}. Store (i, λ_j) in tape $\mathcal{L}_U$.

- $\mathcal{JO}(i, ca)$: It interacts with honest issuer ca. Computes x_i as in $\mathcal{KEO}_u(i)$. Then interacts with ca with x_i. Finally ca returns $cert_{i,ca}$.

- $\mathcal{CO}(i, ca)$: On input the identity, this oracle outputs the user's secret keys. Computes x_i as in $\mathcal{KEO}_u(i)$. Computes (x_{ca}, aux_{ca}) as in $\mathcal{KEO}_c(ca)$. Randomly selects $e \in \mathbb{Z}_p$, and computes:

  $$A = (u/\mathcal{H}_U(i))^{1/(e+x_{ca})}, \quad W = \hat{e}(\mathcal{H}_U(i), g_A).$$

  Stores (i, A, e, W) in reg. Returns (x_i, A, e, aux_{ca}).

  For the ℓ-th query, randomly selects $h \in \mathbb{Z}_p$ and computes $aux_{ca} = g_2^{\gamma} y_A^{-h}$. $\mathcal{S}$ back patch $\mathcal{H}_A(aux_{ca}||ca) = h$. Picks a pair of (A_i, e_i, λ_i) from the k-CAA2 tuple. Back patches (i, λ_i) to $\mathcal{L}_U$. Then we have $x_i = y_U^{\lambda_i}$. Returns $(x_i, A_i, e_i, aux_{ca})$. Computes $W = \hat{e}(\mathcal{H}_U(i), g_A)$. Stores (i, A_i, e_i, W) in reg. Denote this identity as ca_g. If $ca = \mathsf{ca}_g$ in future queries, also runs the above steps.

Other oracles are similar to the proof of theorem 7. ca_g should not be input to the $\mathcal{KEO}_c$. Suppose $\mathcal{A}$ can output a valid signature σ such that the OA cannot trace the identity of the signer, or the OA can find the identity of the signer but cannot prove that to Judge.

Below we proof the soundness of the proof system between Open and Judge. Rewind the simulation to obtain:

$$\begin{aligned} 1 = \Delta Z_1' h^{\Delta z_0'} {t_0'}^{\Delta c'} \quad &\wedge 1 = \hat{e}(h,U)^{\Delta z_0'} {t_1'}^{\Delta c'} \quad \wedge 1 = \hat{e}(h,g_O)^{\Delta z_0'} {t_2'}^{\Delta c'} \\ t_0' = {\Delta Z_1'}^{1/\Delta c'} h^{\Delta z_0'/\Delta c'} \quad &\wedge t_1' = \hat{e}(h,U)^{\Delta z_0'/\Delta c'} \quad \wedge t_2' = \hat{e}(h,g_O)^{\Delta z_0'/\Delta c'} \end{aligned}$$

And notice that we have:

$$\begin{aligned} t_1' &= \hat{e}(h,U)^{s_0'} &&= \hat{e}(t_0',U){m'}^{-1} \\ t_2' &= \hat{e}(h,g_O)^{s_0'} &&= \hat{e}(t_0' {x_{\mathsf{oa}}}^{-1}, g_O) \end{aligned}$$

Let $\tilde{s_0'} = -\Delta z_0'/\Delta c'$. Hence $m' = \hat{e}(t_0',U){t_1'}^{-1} = \hat{e}(h^{\tilde{s_0'}} t_0', U)$. Since we have $t_0' {x_{\mathsf{oa}}}^{-1} = h^{-\tilde{s_0'}}$, then $m' = \hat{e}(h^{\tilde{s_0'}} t_0', U) = \hat{e}(x_{\mathsf{oa}}, U)$. Therefore we extract the witness $x_{\mathsf{oa}} = t_0' h^{\tilde{s_0'}}$. Hence for an OA with secret key x_{oa}, he can always output a valid proof to the Judge if he knows the identity of the signer.

If finally $\mathcal{A}$ returns a signature with group $\mathsf{ca} = \mathsf{ca}_g$, then we rewind the simulation to the point where c is computed.

After rewind, we get: $g_0^{\Delta z_0} t_0^{\Delta c} = 1$, $\Delta Z_1 g_1^{\Delta z_0} t_1^{\Delta c} = 1$, $\Delta Z_2 g_2^{\Delta z_0} t_2^{\Delta c} = 1$, $\Delta Z_3 g_3^{\Delta z_0} t_3^{\Delta c} = 1$, $t_3^{\Delta z_4} t_5^{\Delta c} = 1$, $g_A^{\Delta z_6} U^{\Delta c} = 1$.

Let $\tilde{s}_1 = -\Delta z_0/\Delta c$, $\tilde{x} = \Delta Z_1^{-1/\Delta c}$, $\tilde{\mathcal{H}} = \Delta Z_2^{-1/\Delta c} = \mathcal{H}_1(i)$, $\tilde{A} = \Delta Z_3^{-1/\Delta c}$, $\tilde{e} = -\Delta z_4/\Delta c$, $\tilde{s}_2 = -\Delta z_5/\Delta c$, $\tilde{d} = -\Delta z_6/\Delta c$. We have:

$$\begin{aligned} \hat{e}(g_3,g_A)^{\Delta z_5}[\hat{e}(g_3,S)\hat{e}(g_2,g_A)]^{\Delta z_1} t_6^{\Delta c} &= 1 \\ \hat{e}(g_3,g_A)^{\tilde{s}_2}[\hat{e}(g_3,S)\hat{e}(g_2,g_A)]^{\tilde{s}_1} &= t_6 \\ &= \hat{e}(u,g_A)^{-1}\hat{e}(t_2 t_5, g_A)\hat{e}(t_3,S). \end{aligned}$$

After rearranging, we have:

$$\hat{e}(u,g_A) = \hat{e}(\tilde{A},g_A)^{\tilde{e}} \hat{e}(\tilde{A},S) \hat{e}(\tilde{\mathcal{H}},g_A) e(g_3,g_A)^{\tilde{e}\tilde{s}_1 - \tilde{s}_2}$$

If $\tilde{e}\tilde{s}_1 = \tilde{s}_2$, then we get a pair of $(\tilde{A}, \tilde{e}, \tilde{\mathcal{H}})$ which satisfy $\tilde{A}^{\tilde{e}+\gamma}\tilde{\mathcal{H}} = u$. Then we have $(\tilde{A}, \tilde{e}, \lambda)$, where $(i, \lambda) \in \mathcal{L}_1$, that solves the k-CAA2 problem.

If $\tilde{e}\tilde{s}_1 \neq \tilde{s}_2$, then we have $\tilde{A}^{\tilde{e}+\gamma}\tilde{\mathcal{H}}g_3^{(\tilde{e}\tilde{s}_1-\tilde{s}_2)} = u$. Then we have $\lambda^* = \lambda + \mu(\tilde{e}\tilde{s}_1 - \tilde{s}_2)$, where $(i, \lambda) \in \mathcal{L}_1$, such that $(\tilde{A}, \tilde{e}, \lambda^*)$ solves the k-CAA2 problem.

Hence if $\mathcal{A}$ has a non-negligible advantage ε in winning the game, $\mathcal{S}$ has advantage ε/q_c in solving the k-CAA2 problem.

Now we derive the opposite reduction in the Theorem statement: Give the Adversary a k-CAA2 oracle, and then use it to compute/forge an additional signature, which is not traceable, after k queries to the $\mathcal{CO}$ Oracle. Then $\mathcal{A}$ gets k sets of (A_i, e_i, x_i) for id_i where $1 \leq i \leq k$. $\mathcal{A}$ inputs (A_i, e_i, λ_i), where $(id_i, \lambda_i) \in \mathcal{L}_1$, to the k-CAA2 oracle. The oracle returns a new pair (A_*, e_*, λ_*). $\mathcal{A}$ backpatches (id_*, λ_*) to $\mathcal{L}_1$. $\mathcal{A}$ uses the $\mathcal{KEO}_U$ to find x_* for id_*. Then $\mathcal{A}$ uses (A_*, e_*, x_*) to compute a signature for message m. Then an honest OA will find that the signature is valid and opens to a value $\hat{e}(\mathcal{H}_U(id_*), g_A)$. As it is not in reg, the OA will outputs $\perp$. Hence $\mathcal{A}$ can forge a signature.

Theorem 9 *The IBGS-SDH is non-frameable in the random oracle model if and only if the co-CDH assumption holds.*

Proof. Assume $\mathcal{A}$ can win Experiment NY with advantage ϵ, and it delivers signature σ, message M and a proof ω to signer i_g. It remains to prove that (1) the VE (Verifiable Encryption) part, ω, validly opens to $\hat{e}(\mathcal{H}_U(i_g), g_A)$; and (2) the signature part is sound.

(1) This means if $\mathsf{Judge}(ca, i_g, oa, M, \sigma, \omega) = 1$, then

$$\mathsf{ctxt} = \hat{e}(\mathcal{H}_U(i_g), g_A)\hat{e}(Q_B^{x_O}, U),$$

where $Q_B = \mathcal{H}_O(oa)$. We prove by forking simulation. Some may find this proof approach not rigorous enough. But this is the state-of-the-art proof technique for the correctness of decryption in many results on VE. Besides, it is possible to modify the security model somewhat slightly to make this kind of proof rigorous. We omit details of the modification for the simplicity of presentation.

Suppose $\mathcal{S}$ is given (P, P^α, Q). $\mathcal{S}$ sets $g_U = P, y_U = P^\alpha$ for the identity manager of members.

$\mathcal{S}$ sets $g_A = g_O = P$. $\mathcal{S}$ randomly selects x_A, x_O and computes $y_A = g_A^{x_A}, y_O = g_O^{x_O}$. $\mathcal{S}$ setups the rest of param and provides to $\mathcal{A}$. $\mathcal{S}$ randomly picks ℓ in $\{1, ..., q_H\}$, where q_H is the number of query to $\mathcal{H}_U$.

The oracles are simulated as follows:

- $\mathcal{H}_U(i)$: On input new i, randomly pick $\lambda \in \mathbb{Z}_p$ and return g_U^λ. Store (i, λ) in tape $\mathcal{L}_U$. For the ℓ-th query, return Q and back patch (i, Q) in $\mathcal{L}_U$. Denote this identity as i_g.

- $\mathcal{KEO}_u(i)$: Computes $\mathcal{H}_U(i)$. Then $x_i = y_U^\lambda$, where $(i, \lambda) \in \mathcal{L}_U$. If $i = i_g$, declares failure and exits.

- $\mathcal{CO}(i, ca)$: On input the identity, this oracle outputs the user's secret keys. Computes $\mathcal{H}_1(i)$. Computes $x_i = y_U^\lambda$, where $(i, \lambda) \in \mathcal{L}_U$. Computes $cert_{i,ca}$ as in $\mathcal{IO}(i, ca)$. Returns $(x_i, cert_{i,ca})$. If $i = i_g$, declares failure and exits.

- $\mathcal{SO}(i, ca, M)$: If $i \neq i_g$, computes $x_i, cert_{i,ca}$ as in $\mathcal{CO}$. Then uses $x_i, (A_i, e_i)$ to sign the message M. Return the signature σ.

  If $\mathcal{SO}(i_g, ca, M)$ is called, randomly selects $t_0, ..., t_3, t_5 \in \mathbb{G}_1$, chooses a challenge $c \in \mathbb{Z}_p$ and response $z_0, ..., z_6$ from suitable domains. It computes $\tau_0, ..., \tau_8$ as in GVf. Then back patch c to $\mathcal{H}$ as Eq. 5.2. Obviously this signature will pass GVf.

Finally if $\mathcal{A}$ can frame a member i^* of signing a message m, it has probability $1/q_H$ of framing user i_g. $\mathcal{A}$ should not query $\mathcal{CO}(i^*)$ or $\mathcal{SO}(i^*, m)$. If $i^* = i_g$, $\mathcal{S}$ opens the signature and extracts x_{i_g} as the solution to the co-CDH problem.

(2) This means the soundness of the proof system in Equation 5.1 when ctxt and U are discarded. This is proved in theorem 4.

Hence if $\mathcal{A}$ has a non-negligible advantage ε in winning the game, $\mathcal{S}$ has advantage ε/q_H in solving the co-CDH problem.

Now we derive the opposite reduction in the Theorem statement: Give the Adversary a co-CDH oracle, and then show it can frame. Suppose $\mathcal{A}$ wants to frame user id_*. $\mathcal{A}$ then inputs to the co-CDH oracle: $P = g_U$, $P^\alpha = y_U$, $Q = \mathcal{H}_U(id_*)$. Denote the oracle output Q^α be x_*. Then $\mathcal{A}$ uses x_* to act as an honest user to interact with the Issue Oracle. The oracle outputs a valid certificate (A_*, e_*) for user id_*. Then $\mathcal{A}$ uses (A_*, e_*, x_*) to output a signature σ for message m. After that $\mathcal{A}$ extract the secret key of OA by $\mathcal{KEO}_O$. $\mathcal{A}$ uses it to compute a proof ω that shows id_* signs $sigma$. Hence ω must pass Judge. Then $\mathcal{A}$ outputs $(\sigma, m, id_*, \omega)$ to frame user id_*.

Summarizing, we have:

Theorem 10 *Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be a pairing. The IBGS-SDH is secure if and only if the DDH Assumption in $\mathbb{G}_1$, the co-DBDH Assumption in $(\mathbb{G}_2, \mathbb{G}_1)$, the k-SDH' Assumption, and the co-CDH Assumption all hold in the random oracle model.*

5.5 Discussions

5.5.1 Other Instantiations

For the above generic construction, we use a discrete logarithm type of identity based key pairs for CA and pairing type of identity based key pairs for OA and group members to give an instantiation. From [6], we have three identity based identification for discrete logarithm type: Beth [10], Okamoto [75], BNN [6]. They are suitable for constructing the key pairs for both CA and group members. We have different identity based identification for pairing type ([80], [56], [33]). They are suitable for constructing the key pairs for group members. For OA, the key pairs can be obtained from secure identity based encryption which allows efficient verification. Therefore we can form different identity based group signature using different combination of the above key pairs.

For other kinds of certificates in group signature schemes, CA in Ateniese et al. [2] has private key (p', q') from the strong RSA assumption. However no existing identity based identification has this form of user key pairs. For Dodis et al. [44], there is no CA and the group public key is some accumulated value. Both are not suitable for having identity based group manager.

If one wants the encryption scheme for the open authority to be CCA-2, then we can modify our scheme as follows. We perform the SPK without encryption, and then perform a verifiable encryption scheme from Camenisch and Damgård [26] with Fiat-Shamir heuristic. The encryption scheme used is FullIdent from Boneh and Franklin [18], which is CCA-2. However, the signature size of this scheme will depend on the group size.

5.5.2 Short Ring Signatures

We can formulate our group signature scheme without open authority. We refer this kind of signature scheme as ring signature, as the anonymity of the signature scheme is non-revokeable. It extends the idea of ring signature in [78].

Without the open authority, our signature scheme has signature size independent of the group size. To turn the identity based group signature to a short identity based ring signature scheme, we only have to remove the encryption from GSig. The OA, Open, Judge are also removed. Then short identity based ring signature is constructed.

5.6 Chapter Conclusion

In this chapter, we present a fully identity based group signature scheme, with identity based group manager, identity based group members and identity based open authority. We give a generic construction and also an instantiation, which the signature size is independent of the group size. We prove the security of the instantiation in the random oracle model. We also showed that a short identity based ring signature can be formed similarly.

Chapter 6

Hierarchical IBS without Random Oracles

Hierarchical identity based cryptosystem [50, 58] is a generalization of identity based cryptosystem that mirrors the hierarchy of organizations. An identity at level ℓ of the hierarchy tree can issue private keys to its descendant identities, but cannot sign or decrypt messages for other identities. [50] proposed the idea of Hierarchical Identity Based Signature (HIBS) and Hierarchical Identity Based Encryption (HIBE). In particular, an IBS (resp. IBE) is an 1-level HIBS (resp. HIBE). Combining HIBS and HIBE, [40] proposed the concept of Hierarchical Identity Based SignCryption (HIBSC).

Many reductionist security proofs concerning identity based cryptosystems and other cryptosystems used the random oracle model [8]. Several papers proved that some popular cryptosystems previously proved secure in the random oracle are actually provably insecure when the random oracle is instantiated by any real-world hashing functions [29, 5]. Therefore identity based cryptosystems provably secure in the standard model attract a great interest. [6] showed that a certificate-based IBS is secure without random oracles. However this scheme is less efficient. Several IBE schemes [30, 12, 55] are secure without random oracles under a weaker "selective-

ID" model [30]. Recently, [13] and [90] proposed IBE schemes which are provably secure without random oracles under the stronger model of [18]. [23] proposed an identity based signature without random oracles, but their reduction is tight only if they use the "selective-ID" model.

Most existing practical signature schemes are provably secure in the random oracle model. [49] proposed a variant of hash-and-sign RSA signature scheme, which is provably secure without random oracles, by the strong RSA assumption. A different approach was proposed in [41], and further improvement was proposed in [46]. [21] proposed a signature scheme provably secure under discrete-log type assumption in the standard model, but the signature size is long. [14] proposed a short signature scheme secure without random oracles, under the new q-SDH assumption. [92] proposed some short signatures without random oracles. The signatures originate from the signature schemes in [14, 97, 27, 20]. They showed how these signatures can be constructed and provably secure without random oracles from new assumptions.

It is natural to ask whether other efficient hierarchical identity based cryptosystems are secure without random oracles. In this chapter, we provide an affirmative answer by constructing an HIBS and HIBSC schemes which can be provably secure without random oracles.

Our Contribution

We make the following contributions:

- The *first* constant-size hierarchical identity based signature (HIBS) scheme. It is existentially unforgeable without random oracles under a new interactive intractability assumption.

- Our HIBS scheme is existentially unforgeable providing the Diffie-Hellman Inversion (DHI) Assumption holds in the gauntlet-ID model without

Type	Scheme	Security	ROM	Size
ℓ-HIBS	Cert-chain	Full ACP	No	$O(\ell\lambda_s)$
	This paper	Full ACP/gID-ACP	No	$O(\lambda_s)$
IBS	Cert-chain	Full ACP	No	$O(\lambda_s)$
Standard Signature	[49, 41, 14]	ACP	No	$O(\lambda_s)$

Table 6.1: Recent results on signatures, IBS, and HIBS. Cert-chain: combine *hierarchical authentication tree* and *one-time signatures.* Full ACP: secure against adaptive chosen identity and adaptive chosen message attack. gID-ACP: secure against gauntlet identity and adaptive chosen message attack. ℓ: number of hierarchy level. λ_s: security parameter. ROM: using random oracle model. Our scheme is provably secure in Full ACP or gID-ACP model by using different intractability assumptions.

random oracles. We introduce the gauntlet-ID model, which is a slightly weaker model related to the selective-ID model of [30].

- The *first* constant-size identity based signcryption (IBSC) and hierarchical identity based signcryption (HIBSC) scheme which are provably secure without random oracles.

Our Intuition.

Classic methods of constructing fully secure signatures from combining *hierarchical authentication tree* and *one-time signatures* can be found in [52]. [6] suggested that IBS without random oracles can be constructed by certificate chaining, but it is less efficient. Various instantiations and modifications for IBE are also well-known [31, 19, 17]. We observe that some of these certificate chaining instantiations bear a striking resemblance to the multi-level certificate chaining structure in HIBS. User identity can be certified by his parent, by signing an IBS on the user's identity. The parent's identity can be certified again by one level higher, and the process repeats up until the root.

If in each level, the certification of user identity is secure in the standard model, and finally the lowest level user signature is secure against adaptive chosen message attack in the standard model, then the entire HIBS scheme is Full ACP secure in the standard model. However this solution will increase the signature size by the level of hierarchy. To achieve $O(\lambda_s)$ size HIBS, we need to either use an *interactive intractability assumption*, or lower the security level to gauntlet ID-ACP-UF (which we will define below). We can see that the same case applies for HIBE using sID-CCA. The recent results are summarized in table 6.1, 6.2 and 6.3.

Interactive intractability assumptions: An interactive intractability problem instance means that an attacker can adaptively query an external oracle and can get distinct valid tuples from the oracle which satisfy a relation $\mathcal{R}$. Finally he needs to return a new valid tuple which satisfies $\mathcal{R}$. [65] proposed a LRSW assumption with an external oracle. In proving the security of a signature scheme, the simulator simply forwards all signing oracle queries to this external oracle and returns its output to the adversary. Signature schemes like [27] use this type of assumption. The problem of interactive intractability assumptions is that we need to assume that the tuples return by the oracle should not help the attacker to solve the intractability problem. Therefore we need to be extremely careful when formulating interactive intractability assumptions.

Gauntlet ID-ACP unforgeability: Gauntlet ID-ACP unforgeability means that in the unforgeability game, the adversary finally returns a signature of a user who has never been queried to the key extraction oracle or the signing oracle. It is related to the selective-ID model of [30], which further

Type	Scheme	Security	ROM	Size
ℓ-HIBE	[17] + ?	Full CCA	No	$O(\ell\lambda_s)$
	[17] + [15]	sID-CCA	No	$O(\lambda_s)$
IBE	[90]	Full CCA	No	$O(\lambda_s)$
Standard Encryption	Cramer-Shoup/OAEP/ [17]+sID-CCA IBE	CCA	No	$O(\lambda_s)$

Table 6.2: Recent results on encryptions, IBE, and HIBE. Full CCA: secure against adaptive chosen identity and adaptive chosen ciphertext attack. sID-CCA: secure against selective identity and adaptive chosen ciphertext attack. ℓ: number of hierarchy level. λ_s: security parameter. ROM: using random oracle model. The first row means that full CCA secure HIBE can be achieved by using [17] and an adaptive chosen identity and chosen plaintext secure HIBE. However no existing scheme achieves this with a tight security reduction.

requires the adversary to select the user he attacks at the beginning of the unforgeability game.

We observe that by using either approach, we can achieve a constant size HIBS secure without random oracles.

6.1 New Intractability Assumption

We introduce a new intractability assumption called the OrcYW assumption that will be used later.

Definition 38 *The* OrcYW Problem *is that given*

1. $\ell \geq 1$, $\{g^{x^i} : 0 \leq i \leq \ell\}$, γ, δ, g_4, g_5, γ_1, $\cdots$, γ_ℓ, *an identity* $I = \{I_1, \cdots, I_\ell\}$, *full-domain collision-resistant hash function* $\mathcal{H}$,

2. *an oracle* $O_{\mathcal{H}}$ *which upon input a message* m *and an identity* $I' = \{I_1, \cdots, I_k\}$ *for* $k \leq \ell$, *outputs a tuple* (D_1, D_2, Z_1, Z_2) *satisfying: For some*

Type	Scheme	Security	ROM	Size
ℓ-HIBSC	[40]	Full CCA + ACP	Yes	$O(\ell\lambda_s)$
	This paper	sID-CCA + Full ACP	No	$O(\lambda_s)$
IBSC	[22], [96], etc.	Full CCA + ACP	Yes	$O(\lambda_s)$
	This paper	sID-CCA + Full ACP	No	$O(\lambda_s)$
Standard Signcryption	[1], [43]	CCA + ACP	No	$O(\lambda_s)$

Table 6.3: Recent results on signcryption, IBSC, and HIBSC. All notations are defined in table 6.1 and 6.2. [43] showed that only standard signcryption scheme of [1] and [43] achieves the strong *insider* security model. All existing IBSC and HIBSC schemes are provably secure in the random oracles only.

random t, r, which differ for each query to $O_{\mathcal{H}}$,

$$D_1 = g^t, \quad D_2 = Q^t, \quad Z_1 = a_0^h g_4^t, \quad Z_2 = a_1^h g_5^t$$

where

$$\begin{aligned} Q &= g_3 \prod_{i=1}^{k} h_i^{I_i}, \quad h_i = g^{\gamma_i} g^{-x^{\ell-i+1}}, \textit{ for } 1 \le i \le \ell \\ g_2 &= g^{x^\ell+\gamma}, \quad g_3 = g^{\delta+\sum_{i=1}^{\ell} x^{\ell-i+1} I_i}, \\ a_0 &= g_2^x Q^r, \quad a_1 = g^r, \quad h = \mathcal{H}(D_1, D_2, I', m, \mathsf{param}), \\ \mathsf{param} &= (g, g^x, g_2, g_3, g_4, g_5, h_1, \cdots, h_\ell) \end{aligned}$$

to output $(\tilde{m}, \tilde{D}_1, \tilde{D}_2, \tilde{Z}_1, \tilde{Z}_2)$ satisfying

$$\begin{aligned} & \hat{e}(g, \tilde{Z}_1) \cdot \hat{e}(g_5, \tilde{D}_2) = \hat{e}(g_1, g_2)^{\tilde{h}} \cdot \hat{e}(\tilde{D}_1, g_4) \cdot \hat{e}(\tilde{Z}_2, Q) \\ \wedge \ & \hat{e}(\tilde{D}_1, Q) = \hat{e}(g, \tilde{D}_2) \ \wedge \ \tilde{m} \textit{ was not queried to } O_{\mathcal{H}} \\ \wedge \ & Q = g_3 \prod_{i=1}^{\ell} h_i^{I_i} \end{aligned}$$

where $\tilde{h} = \mathcal{H}(\tilde{D}_1, \tilde{D}_2, I, \tilde{m}, \mathsf{param})$.

We say that the *OrcYW Assumption* holds if no PPT algorithm can solve a random instance of the OrcYW Problem with non-negligible probability.

The intractability of the OrcYW Assumption will be discussed in section 6.4.

6.2 Security Model: HIBS and HIBSC

We present the security models for HIBS (Hierarchical Identity Based Signatures) and for HIBSC (Hierarchical Identity Based Signcryption).

6.2.1 HIBS Security Model

In identity based cryptography, the security model for IBE was proposed in [18]. Besides the decryption oracle, the adversary is also allowed to query the key extraction oracle adaptively to extract the secret key for any identity except the challenge identity. [30] proposed a weaker "selective-identity" model, where the adversary selects the challenge identity in advance, before the public parameter is generated. In this paper, we will introduce a variant for signature scheme, namely a "gauntlet-identity" model.

An ℓ-level HIBS scheme consists of four algorithms: (Setup, Der, Sign, Verify). The algorithms are specified as follows:

- Setup: On input a security parameter 1^{λ_s}, the TA generates $\langle \mathsf{msk}, \mathsf{param} \rangle$ where msk is the randomly generated master secret key, and param is the corresponding public parameter.
- Der: On input an identity vector ID, its associated secret key SK_{ID}, and a string r, it returns the corresponding private key $SK_{\mathsf{ID}.r}$ (corresponds to param).

- Sign: On input the private key of the signer ID, SK_{ID} and a message M, it outputs a signature σ corresponding to param.

- Verify: On input the signer identity vector ID, a message M and signature σ, it outputs $\top$ if σ is a valid signature of M corresponding to $\mathsf{ID}, \mathsf{param}$. Otherwise, it outputs $\perp$.

The security of a HIBS consists of two requirements, namely *Correctness* and *Existential Unforgeability.* They are defined as follows:

Correctness.

We require that $\top \leftarrow \mathsf{Verify}(\mathsf{ID}, M, \mathsf{Sign}(SK_{\mathsf{ID}}, M))$ for any message M, any private key SK_{ID} and its corresponding identity ID.

Existential Unforgeability.

We define the existential unforgeability against adaptive identity and adaptive chosen plaintext attack for HIBS (ACP-UF). We require that the user identity should be queried through an oracle as in [6]. We assume the simulator maintains an honest user list HU and a corrupt user list CU. We define the following oracles:

- $\mathcal{IO}(\mathsf{ID})$: The Initialization Oracle with input ID outputs $\perp$ if $\mathsf{ID} \in HU \cup CU$. If ID's prefix is in CU, it puts ID in CU and returns 1. Otherwise it puts ID in HU and returns 1.

- $\mathcal{KEO}(\mathsf{ID})$: The Key Extraction Oracle with input ID outputs $\perp$ if $\mathsf{ID} \notin HU$. Otherwise it outputs the corresponding secret key SK_{ID}, removes ID and its children from HU and adds them to CU.

- $\mathcal{SO}(\mathsf{ID}, M)$: The Signing Oracle with input signer ID and message M outputs $\perp$ if $\mathsf{ID} \notin HU$. Otherwise it will output a signature σ such that $\mathsf{Verify}(\mathsf{ID}, M, \sigma) = \top$.

The Game is defined as follows:

1. (*Init. Phase*) Simulator $\mathcal{S}$ generates system parameter param and gives it to Adversary $\mathcal{A}$.

2. (*Probe Phase*) $\mathcal{A}$ queries $\mathcal{IO}(\mathsf{ID})$, $\mathcal{KEO}(\mathsf{ID})$ and $\mathcal{SO}(\mathsf{ID}, M)$, in arbitrary interleaf.

3. (*End Game*) $\mathcal{A}$ delivers a signature σ_{ga} for signer identity ID_{ga} and message M_{ga}. ID_{ga} or its prefix have never been input to a $\mathcal{KEO}$ and σ_{ga} should not be the output of $\mathcal{SO}(\mathsf{ID}_{ga}, M_{ga})$.

$\mathcal{A}$ *wins* if he completes the Game with $\top = \mathsf{Verify}(\mathsf{ID}_{ga}, M_{ga}, \sigma_{ga})$ and $\mathsf{ID}_{ga} \in HU$. Its *advantage* is its probability of winning.

Definition 39 *The HIBS scheme is* ACP-UF *secure if no PPT adversary $\mathcal{A}$ has non-negligible advantage in the ACP-UF game.*

We say that a HIBS is *secure* if it satisfies *Correctness* and *Existential Unforgeability.*

Gauntlet-ID Existential Unforgeability.

We define the existential unforgeability against gauntlet identity and adaptive chosen plaintext attack for HIBS (gID-ACP-UF) as follows. The game is similar to the ACP-UF game, except in the end game phase, ID_{ga} or its prefix have never been input to a $\mathcal{SO}$ query. The HIBS scheme is *gID-ACP-UF* secure if no PPT adversary $\mathcal{A}$ has non-negligible advantage in the gID-ACP-UF game.

Remark: [33] and many IBS schemes have the $\mathcal{SO}$ query of the gauntlet ID to be handled by the random oracle. They also disallow the query of gauntlet ID to the $\mathcal{KEO}$, which is similar to our gID model.

Selective-ID Existential Unforgeability.

We define the existential unforgeability against selective identity and adaptive chosen plaintext attack for HIBS (sID-ACP-UF) as follows. The game is similar to the gID-ACP-UF game, except before the Init. phase, $\mathcal{A}$ gives ID_{ga} to $\mathcal{S}$ in advance. The HIBS scheme is *sID-ACP-UF* secure if no PPT adversary $\mathcal{A}$ has non-negligible advantage in the sID-ACP-UF game.

6.2.2 Hierarchical Identity Based Signcryption (HIBSC)

An ℓ-level HIBSC scheme consists of four algorithms: (Setup, Der, $\mathsf{Signcrypt}$, $\mathsf{Unsigncrypt}$). The algorithms are specified as follows:

- Setup: On input a security parameter 1^{λ_s}, the TA generates $\langle \mathsf{msk}_A$, msk_B, $\mathsf{param}\rangle$ where msk_A (resp. msk_B) is the randomly generated master secret key for signcryptor (resp. unsigncryptor), and param is the corresponding public parameter.

- Der: On input an identity vector ID, its associated secret key SK_{ID}, and a string r, it returns the corresponding private key $SK_{\mathsf{ID}.r}$ (corresponds to param).

- $\mathsf{Signcrypt}$: On input the private key of the signer ID_A, SK_{ID_A}, the recipient identity ID_B and a message M, it outputs a ciphertext σ corresponding to param.

- $\mathsf{Unsigncrypt}$: On input the private key of the recipient ID_B, SK_{ID_B}, and a signature σ, it decrypts to a message M, sender identity ID_A and a signature s. It outputs M and ID_A if s is valid corresponding to $M, \mathsf{ID}_A, \mathsf{ID}_B, \mathsf{param}$ and signer = encryptor. Otherwise, it outputs $\perp$.

The security of a HIBSC consists of three requirements, namely *Correctness*, *Indistinguishability* and *Existential Unforgeability*. They are defined as follows:

Correctness.

We require that $M \leftarrow \mathsf{Unsigncrypt}(SK_{\mathsf{ID}_B}, \mathsf{Signcrypt}(SK_{\mathsf{ID}_A}, \mathsf{ID}_B, M))$ for any message M, any private key SK_{ID} and its corresponding identity ID.

Indistinguishability.

We define the indistinguishability against selective identity and adaptive chosen ciphertext attack for HIBS (sID-IND-CCA), as in the following game. We define the following oracles:

- $\mathcal{KEO}_{A/B}(\mathsf{ID})$: The Key Extraction Oracle with input ID will output the secret key SK_{ID} corresponding to msk_A or msk_B.
- $\mathcal{SCO}(\mathsf{ID}_A, \mathsf{ID}_B, M)$: The Signcryption Oracle with input signer identity ID_A, recipient identity ID_B and message M will output a ciphertext σ such that $\mathsf{Unsigncrypt}(SK_{\mathsf{ID}_B}, \sigma) = M$.
- $\mathcal{UO}(\mathsf{ID}_B, \sigma)$: The Unsigncryption Oracle with input recipient identity ID_B and ciphertext M will output a message M and the sender identity ID_A for a valid ciphertext σ or will output $\perp$ otherwise.

The Game is defined as follows:

1. (*Setup Phase*) Adversary $\mathcal{A}$ gives recipient ID_B^* to Simulator $\mathcal{S}$. Then $\mathcal{S}$ generates system parameter param and gives $(\mathsf{param}, \mathsf{msk}_A)$ to Adversary $\mathcal{A}$.
2. (*Probe 1 Phase*) $\mathcal{A}$ queries $\mathcal{KEO}_B$, $\mathcal{SCO}$, and $\mathcal{UO}$ in arbitrary interleaf.

3. (*Gauntlet Phase*) $\mathcal{A}$ gives two messages M_0^*, M_1^* and sender ID_A^* to $\mathcal{S}$. $\mathcal{S}$ randomly picks a bit b and returns $\sigma^* = \mathsf{Signcrypt}(SK_{\mathsf{ID}_A^*}, \mathsf{ID}_B^*, M_b^*)$ to $\mathcal{A}$.

4. (*Probe 2 Phase*) $\mathcal{A}$ queries $\mathcal{KEO}_B$, $\mathcal{SCO}$, and $\mathcal{UO}$ in arbitrary interleaf.

5. (*End Game*) $\mathcal{A}$ delivers a guess $\hat{b}$.

$\mathcal{A}$ *wins* if the following holds: $\hat{b} = b$ and ID_B^* or its prefix has never been queried to the $\mathcal{KEO}_B$ and $(\mathsf{ID}_B^*, \sigma^*)$ has never been queried to the $\mathcal{UO}$. $\mathcal{A}$'s *advantage* is its probability that he wins over half. The HIBSC is *sID-IND-CCA* secure if no PPT attacker has a non-negligible advantage in the Indistinguishability Game.

Existential Unforgeability.

We define the existential unforgeability against adaptive chosen identity and adaptive chosen plaintext attack for HIBSC (ACP-UF), as in the following game.

1. (*Setup Phase*) $\mathcal{S}$ sets up system parameters and gives $(\mathsf{param}, \mathsf{msk}_B)$ to Adversary $\mathcal{A}$.

2. (*Probe Phase*) $\mathcal{A}$ queries $\mathcal{KEO}_A$, $\mathcal{SCO}$, and $\mathcal{UO}$ in arbitrary interleaf.

3. (*End Game*) $\mathcal{A}$ delivers a ciphertext σ^* and a recipient identity ID_B^*.

$\mathcal{A}$ *wins* if the following holds: $(M^*, \mathsf{ID}_A^*) \leftarrow \mathsf{Unsigncrypt}(\sigma^*, SK_{\mathsf{ID}_B^*})$, ID_A^* or its prefix has never been queried to the $\mathcal{KEO}_A$ and no $\mathcal{SO}$ request has resulted in a ciphertext C_i, whose unsigncryption under $SK_{\mathsf{ID}_B^*}$ is identical to the triple $(M^*, \mathsf{ID}_A^*, \sigma^*)$. $\mathcal{A}$'s *advantage* is the probability that he wins. The HIBSC is *ACP-UF* secure if no PPT attacker has a non-negligible advantage in the Unforgeability Game.

We say that a HIBSC is *secure* if it satisfies *Correctness*, *Indistinguishability* and *Existential Unforgeability*.

6.3 Efficient Instantiation of HIBS

We construct an efficient ℓ-level HIBS scheme which is provably secure without random oracles, based on the ℓ-DHI* assumption. The key system comes from [15].

Let $\mathbb{G}$ be a bilinear group of prime order p. Given a pairing: $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.

Setup: To generate system parameters, the algorithm selects a random generator g, g_2, g_3, g_4, g_5, h_1, $\ldots$, $h_\ell \in \mathbb{G}$, picks a random $\alpha \in \mathbb{Z}_p$, and sets $g_1 = g^\alpha$. It chooses an collision-resistant hash function $\mathcal{H}$. Note $\mathcal{H}$ is not a random oracle. Anyone, including the attacker, can compute $\mathcal{H}$ in private. The system parameters $\mathsf{param} = (g, g_1, g_2, g_3, g_4, g_5, h_1, \ldots, h_\ell, \mathcal{H})$ and the master key is g_2^α.

Der: To generate a private key for $\mathsf{ID} = (\mathsf{id}_1, \ldots, \mathsf{id}_k)$. where $k \leq \ell$, the algorithm picks a random $r \in \mathbb{Z}_p^*$ and computes:

$$SK_{\mathsf{ID}} = \left(g_2^\alpha Q_{\mathsf{ID}}^r,\ \ g^r,\ \ h_{k+1}^r, \ldots, h_\ell^r\right) = (a_0, a_1, b_{k+1}, \ldots, b_\ell)$$

where $Q_{\mathsf{ID}} = h_1^{\mathsf{id}_1} \cdots h_k^{\mathsf{id}_k} \cdot g_3$. The private key for ID can also be generated by its parent $\mathsf{ID}_{|k-1} = (\mathsf{id}_1, \ldots, \mathsf{id}_{k-1})$. Details refer to [15].

Sign: For a user with identity ID and private key SK_{ID}, he signs a message M as follows. He picks random $t, \bar{r} \in \mathbb{Z}_p$, and computes:

$$D_1 = g^t,\ \ D_2 = {Q_{\mathsf{ID}}}^t,\ \ h = H(D_1, D_2, \mathsf{ID}, M, \mathsf{param})$$
$$\bar{a}_0 = a_0 {Q_{\mathsf{ID}}}^{\bar{r}},\ \ \bar{a}_1 = a_1 g^{\bar{r}},\ \ Z_1 = {\bar{a}_0}^h g_4^t,\ \ Z_2 = {\bar{a}_1}^h g_5^t$$

The signature σ is (D_1, D_2, Z_1, Z_2).

Verify: The verifier receives a signature $\sigma = (D_1, D_2, Z_1, Z_2)$ for message M and signer ID, he computes $h = H(D_1, D_2, \mathsf{ID}, M, \mathsf{param})$. The verifier checks if both of the following relations hold:

$$\hat{e}(g, Z_1) \cdot \hat{e}(g_5, D_2) \stackrel{?}{=} \hat{e}(g_1, g_2)^h \cdot \hat{e}(D_1, g_4) \cdot \hat{e}(Z_2, Q_{\mathsf{ID}})$$
$$\hat{e}(D_1, Q_{\mathsf{ID}}) \stackrel{?}{=} \hat{e}(g, D_2)$$

The verifier outputs $\top$ if it is true. Otherwise, he outputs $\perp$.

Remark: We can view Q_{ID} as the output of a hash function with input ID. In many HIBE schemes like [13, 90, 15], they specify $Q_{\mathsf{ID}} = h_1^{\mathsf{id}_1} \cdots h_k^{\mathsf{id}_k} \cdot g_3$.

6.3.1 Security Analysis

We will prove the security of the HIBS scheme using the new OrcYW assumption and other assumptions.

Theorem 11 *Assume $\mathcal{H}$ is a full-domain collision-resistant hash function. The hierarchical identity based signature scheme* $\mathsf{HIBS}_{\mathsf{BBG}}(\ell)$ *is* correct *and* ACP-UF *secure provided the* OrcYW Assumption *holds.*

Proof. The correctness of the scheme is shown as follows:

$$\begin{aligned}
& \hat{e}(g, Z_1) \cdot \hat{e}(g_5, D_2) \\
= \ & \hat{e}(g, g_2^{\alpha} \cdot (h_1^{\mathsf{id}_1} \cdots h_k^{\mathsf{id}_k} \cdot g_3)^{r+\bar{r}})^h \cdot \hat{e}(g, g_4)^t \cdot \hat{e}(g_5, h_1^{\mathsf{id}_1} \cdots h_k^{\mathsf{id}_k} \cdot g_3)^t \\
= \ & \hat{e}(g^{\alpha}, g_2)^h \cdot \hat{e}(g^{r+\bar{r}}, h_1^{\mathsf{id}_1} \cdots h_k^{\mathsf{id}_k} \cdot g_3)^h \cdot \hat{e}(D_1, g_4) \cdot \hat{e}(g_5^t, h_1^{\mathsf{id}_1} \cdots h_k^{\mathsf{id}_k} \cdot g_3) \\
= \ & \hat{e}(g_1, g_2)^h \cdot \hat{e}(D_1, g_4) \cdot \hat{e}(Z_2, h_1^{\mathsf{id}_1} \cdots h_k^{\mathsf{id}_k} \cdot g_3)
\end{aligned}$$

Next, we prove ACP-UF.

Setup:

Simulator $\mathcal{S}$ received a OrcYW Problem instance: $\{g^{x^i} : 0 \leq i \leq \ell\}$,$\gamma$, δ, g_4, g_5, γ_1, $\cdots$, γ_ℓ, a special identity chain $\mathbf{I}^* = \{I_1^*, \ldots, I_\ell^*\}$, a full-domain collision-resistant hash function $\mathcal{H}$ and an oracle O_{YW}.

$\mathcal{S}$ computes $g_1 = g^x$, $g_2 = g^{x^\ell + \gamma}$, $g_3 = g^{\delta + \sum_{j=1}^{\ell} x^{\ell-j+1} I_j^*}$ and $h_j = g^{\gamma_j} g^{-x^{\ell-j+1}}$, for $1 \leq j \leq \ell$. $\mathcal{S}$ initializes two empty list HU and CU. $\mathcal{S}$ randomly selects n identity chains $\mathbf{I}_1, \ldots, \mathbf{I}_n$ and puts them in HU, including $\mathbf{I}^*$ in it. $\mathcal{S}$ gives the public parameters $\mathsf{param} = (g, g_1, g_2, g_3, g_4, g_5, h_1, \ldots, h_\ell)$ and n identity chains to $\mathcal{A}$.

Simulating $\mathcal{IO}$:

For input ID, $\mathcal{S}$ outputs $\perp$ if $\mathsf{ID} \in HU \cup CU$. If ID's prefix is in CU, it puts ID in CU and returns 1. Otherwise it puts ID in HU and returns 1.

Simulating $\mathcal{KEO}$:

Simulate as in [15]. For input identity $\mathsf{ID} = (\mathsf{id}_1, \ldots, \mathsf{id}_u)$, if ID is $\mathbf{I}^*$ or a prefix of it, the simulator declares failure and exits. If $\mathsf{ID} \notin HU$, $\mathcal{S}$ outputs $\perp$. Otherwise there exists a $k \leq u$ such that $\mathsf{id}_k \neq I_k^*$. We set k be the smallest such index. To answer the query, the simulator derives a secret key for the identity $(\mathsf{id}_1, \ldots, \mathsf{id}_k)$ from which it then constructs a private key for $\mathsf{ID} = (\mathsf{id}_1, \ldots, \mathsf{id}_k, \ldots, \mathsf{id}_u)$.

To generate the secret key for the identity $(\mathsf{id}_1, \ldots, \mathsf{id}_k)$, the simulator chooses a random $\tilde{r} \in \mathbb{Z}_p$. Denote $r = \frac{x^k}{(\mathsf{id}_k - I_k^*)} + \tilde{r}$ and compute:

$$\begin{aligned} a_0 &= y_1^\gamma \cdot Z \cdot g^{x^{\ell-k+1}\tilde{r}(I_k^* - \mathsf{id}_k)} \text{ where } Z = \left(g^{\delta + \sum_{i=1}^{k} \mathsf{id}_i \gamma_i} \cdot \prod_{i=k+1}^{\ell} g^{x^{\ell-i+1} I_i^*} \right)^r \\ a_1 &= g^r = g^{x^k/(\mathsf{id}_k - I_k^*)} g^{\tilde{r}} \end{aligned}$$

Refer to [15] for the well-formedness of the secret key. The remaining h_{k+1}^r,

$\ldots$, h_ℓ^r can be computed by the simulator since they do not involve a $g^{x^{\ell+1}}$ term. Finally $\mathcal{S}$ removes ID and its children in HU and puts them in CU.

Simulating $\mathcal{SO}$:

For query with $(\mathsf{ID}_\tau, m_\tau)$, if $\mathsf{ID} \notin HU$, $\mathcal{S}$ outputs $\perp$. If ID_τ is $\mathbf{I}^*$ or its prefix, $\mathcal{S}$ queries $O_{YW}(m_\tau, \mathsf{ID}_\tau)$ and forwards the answer to $\mathcal{A}$.

Otherwise, $\mathcal{S}$ computes the secret key of ID_τ as in $\mathcal{KEO}$, and then computes the signature using the secret key.

Simulation Deviation:

It can be shown that the statistical distance among the Real World and the Ideal World is negligible.

Extraction:

$\mathcal{A}$ outputs $(D_1^*, D_2^*, Z_1^*, Z_2^*)$ for signer $I^* \in \{\mathbf{I}_1, \ldots, \mathbf{I}_n\}$ and message m^*. If $I^* \neq \mathbf{I}^*$, $\mathcal{S}$ declares failure and exits. Otherwise, as (I^*, m^*) has never been queried to $\mathcal{SO}$, $\mathcal{S}$ can use the signature to answer the problem instance.

We also prove the security of our scheme in the gauntlet-ID model, using an intractability assumption without interactive oracle.

Theorem 12 *Assume $\mathcal{H}$ is a full-domain collision-resistant hash function. The hierarchical identity based signature scheme* $\mathsf{HIBS}_{\mathsf{BBG}}(\ell)$ *is* correct *and* gID-ACP-UF *secure provided the* ℓ-DHI* Assumption *holds.*

Proof. Suppose a simulator $\mathcal{S}$ is given the ℓ-DHI* tuple $(g, g^x, \ldots, g^{x^\ell})$. $\mathcal{S}$ initializes two empty list HU and CU. The gID-ACP-UF games begins with a simulator randomly picks $\mathcal{S}$ randomly selects n identity chains $\tilde{\mathsf{ID}} =$

$\{\mathbf{I}_1, \ldots, \mathbf{I}_n\}$ and puts them in HU. Denote $\mathbf{I}^* = \{I_1^*, \ldots, I_\ell^*\}$ be an identity in $\tilde{\mathsf{ID}}$.

The simulator picks a random $\gamma \in \mathbb{Z}_p$ and assigns $g_1 = g^x, g_2 = g^{x^\ell} \cdot g^\gamma$. The simulator picks random $\gamma_1, \ldots \gamma_\ell \in \mathbb{Z}_p$ and sets $h_j = g^{\gamma_j} g^{-x^{\ell-j+1}}$, for $1 \leq j \leq \ell$. It also picks a random $\delta \in \mathbb{Z}_p$ and computes $g_3 = g^{\delta + \sum_{j=1}^{\ell} x^{\ell-j+1} I_j^*}$. The simulator picks random $\omega_1, \omega_2 \in \mathbb{Z}_p$ and sets $g_4 = g^{\omega_1}, g_5 = g^{\omega_2}$. The simulator gives the adversary $\mathcal{A}$ the public parameters $\mathsf{param} = (g,\ g_1,\ g_2,\ g_3,\ g_4,\ g_5,\ h_1,\ \ldots,\ h_\ell)$ and $\tilde{\mathsf{ID}}$. The corresponding (unknown) master secret key is $g_2^x = g^{x(x^\ell + \gamma)}$.

Simulating $\mathcal{IO}$:

For input ID, $\mathcal{S}$ outputs $\perp$ if $\mathsf{ID} \in HU \cup CU$. If ID's prefix is in CU, it puts ID in CU and returns 1. Otherwise it puts ID in HU and returns 1.

Simulating $\mathcal{KEO}$:

Simulate as in [15]. For input identity $\mathsf{ID} = (\mathsf{id}_1, \ldots, \mathsf{id}_u)$, if ID is $\mathbf{I}^*$ or a prefix of it, the simulator declares failure and exits. If $\mathsf{ID} \notin HU$, $\mathcal{S}$ outputs $\perp$. Otherwise there exists a $k \leq u$ such that $\mathsf{id}_k \neq I_k^*$. We set k be the smallest such index. To answer the query, the simulator derives a secret key for the identity $(\mathsf{id}_1, \ldots, \mathsf{id}_k)$ from which it then constructs a private key for $\mathsf{ID} = (\mathsf{id}_1, \ldots, \mathsf{id}_k, \ldots, \mathsf{id}_u)$.

To generate the secret key for the identity $(\mathsf{id}_1, \ldots, \mathsf{id}_k)$, the simulator chooses a random $\tilde{r} \in \mathbb{Z}_p$. Denote $r = \frac{x^k}{(\mathsf{id}_k - I_k^*)} + \tilde{r}$ and compute:

$$\begin{aligned} a_0 &= y_1^\gamma \cdot Z \cdot g^{x^{\ell-k+1} \tilde{r} (I_k^* - \mathsf{id}_k)} \quad \text{where } Z = \left(g^{\delta + \sum_{i=1}^{k} \mathsf{id}_i \gamma_i} \cdot \prod_{i=k+1}^{\ell} g^{x^{\ell-i+1} I_i^*} \right)^r \\ a_1 &= g^r = g^{x^k / (\mathsf{id}_k - I_k^*)} g^{\tilde{r}} \end{aligned}$$

Refer to [15] for the well-formedness of the secret key. The remaining h_{k+1}^r,

..., h_ℓ^r can be computed by the simulator since they do not involve a $g^{x^{\ell+1}}$ term. Finally $\mathcal{S}$ removes ID and its children in HU and puts them in CU.

Simulating $\mathcal{SO}$:

For query with $(\mathsf{ID}_\tau, m_\tau)$, if $\mathsf{ID} \notin HU$, $\mathcal{S}$ outputs $\perp$. If ID_τ is $\mathbf{I}^*$ or its prefix, the simulator declares failure and exits. Otherwise, $\mathcal{S}$ computes the secret key of ID_τ as in $\mathcal{KEO}$, and then computes the signature using the secret key.

Simulation Deviation:

It can be shown that the statistical distance among the Real World and the Ideal World is negligible.

Extraction:

Finally, the adversary $\mathcal{A}$ returns a signature σ^* for message M^* and signer $\hat{ID} \in \tilde{\mathsf{ID}}$, where $\hat{ID}$ or its prefix is never been queried to $\mathcal{KEO}$ or $\mathcal{SO}$. For probability $1/n_1$, $\hat{ID} = \mathbf{I}^*$. Otherwise $\mathcal{S}$ declares failure and exits. We denote $\sigma^* = (D_1^*, D_2^*, Z_1^*, Z_2^*)$. Then we compute $h = H(D_1^*, D_2^*, \mathbf{I}^*, M^*, \mathsf{param})$ and we have:

$$D_1^* = g^t \qquad Z_1^* = a_0^h g_4^t = a_0^h g^{\omega_1 t} \qquad Z_2^* = a_1^h g_5^t = a_1^h g^{\omega_2 t}$$

Then we can compute $a_0 = (Z_1^*/D_1^{*\omega_1})^{1/h}$ and $a_1 = (Z_2^*/D_1^{*\omega_2})^{1/h}$. Therefore for $\mathbf{I}^*$, we can set $a_1 = g^{\bar{r}}$ for some $\bar{r} \in \mathbb{Z}_p$. Then:

$$\begin{array}{rcl} a_0 & = & g_2^\alpha (g_3 \prod_{i=1}^{\ell} h_i^{I_i^*})^{\bar{r}} \\ & = & g_2^\alpha (g^{\delta + \sum_{i=1}^{\ell} (\gamma_i I_i^*)})^{\bar{r}} \end{array}$$

Therefore the simulator returns $g^{x^{\ell+1}} = g_2^\alpha / g^{x\gamma} = a_0 / (a_1^{\delta + \sum_{i=1}^{\ell}(\gamma_i I_i^*)} g^{x\gamma})$ as the solution.

6.3.2 Ordinary Signature from HIBS

For a secure HIBS scheme with $\ell = 1$, we obtain a secure IBS scheme. We further show that we have a secure (ordinary) signature scheme from a secure IBS scheme. The construction is as follows:

(Ordinary) signature scheme:

- **Setup:** The user secret key sk is the master key of IBS. The user public key pk is param of IBS.
- **Sign:** The signer picks random β and generates the secret key sk_β for its child β as in IBS. The signer uses sk_β to sign the message M as in IBS. The signature is the IBS signature plus β.
- **Verify:** The verifier checks the validity of the IBS signature with respect to identity β and pk.

We give our instantiation as follows:

(Ordinary) signature scheme $\mathsf{Sig}_{\mathsf{IBS.BBG}}$**:**

- **Setup:** To generate system parameters, the algorithm selects a random generator g, g_2, g_3, g_4, g_5, $h_1 \in \mathbb{G}$, picks a random $\alpha \in \mathbb{Z}_p$, and sets $g_1 = g^\alpha$. It chooses an collision-resistant hash function $\mathcal{H}$. The public keys are $pk = (g, g_1, g_2, g_3, g_4, g_5, h_1, \mathcal{H})$ and the secret key is g_2^α.
- **Sign:** The signer picks random $t, \beta, \bar{r} \in \mathbb{Z}_p$, and computes:

$$D_1 = g^t, \quad D_2 = \left(g_3 h_1^\beta\right)^t, \quad h = H(D_1, D_2, \beta, M, pk)$$
$$\bar{a}_0 = g_2^\alpha \left(g_3 h_1^\beta\right)^{\bar{r}}, \quad \bar{a}_1 = g^{\bar{r}}, \quad Z_1 = {\bar{a}_0}^h g_4^t, \quad Z_2 = {\bar{a}_1}^h g_5^t$$

The signature σ is $(D_1, D_2, Z_1, Z_2, \beta)$.

- **Verify:** The verifier receives a signature σ $=(D_1, D_2, Z_1, Z_2, \beta)$ for message M, he computes $h = H(D_1, D_2, \beta, M, pk)$. The verifier checks if both of the following relations hold:

$$\hat{e}(g, Z_1) \cdot \hat{e}(g_5, D_2) \overset{?}{=} \hat{e}(g_1, g_2)^h \cdot \hat{e}(D_1, g_4) \cdot \hat{e}(Z_2, g_3 h_1^{\beta})$$
$$\hat{e}(D_1, g_3 h_1^{\beta}) \overset{?}{=} \hat{e}(g, D_2)$$

The verifier outputs $\top$ if it is true. Otherwise, he outputs $\perp$.

6.4 Plausibility Arguments for the Intractability of the OrcYW Assumption

Assuming knowledge of $\omega_1 = \log_g g_4$ *and* $\omega_2 = \log_g g_5$, then an OrcYW Problem solver can solve the DHI* Problem. But $\mathcal{S}$ can also solve the DHI* Problem from one OrcYW query outcome, using this knowledge. Let $\tilde{t} = \log_g \tilde{D}_1$, $\tilde{t}' = \log_Q \tilde{D}_2$. Then the relation:

$$\hat{e}(\tilde{D}_1, Q) = \hat{e}(g, \tilde{D}_2)$$

implies $\tilde{t} = \tilde{t}'$. Let A, B be such that $\tilde{Z}_1 = g_2^{x\tilde{h}} A g_4^{\tilde{t}}$, $\tilde{Z}_2 = B g_5^{\tilde{t}}$. Let $\tilde{r} = (\tilde{h})^{-1} \log_Q A$, $\tilde{r}' = (\tilde{h})^{-1} \log_g B$. Then the relation:

$$\hat{e}(g, \tilde{Z}_1) \cdot \hat{e}(g_5, \tilde{D}_2) = \hat{e}(g_1, g_2)^{\tilde{h}} \cdot \hat{e}(\tilde{D}_1, g_4) \cdot \hat{e}(\tilde{Z}_2, Q)$$

implies $\tilde{r} = \tilde{r}'$. Finally, $\mathcal{S}$ computes $\bar{A} = g_2^x Q^{\tilde{r}} = (\tilde{Z}_1 g_4^{-\tilde{t}})^{1/\tilde{h}} = (\tilde{Z}_1 (\tilde{D}_1)^{-\omega_1})^{1/\tilde{h}}$, $\bar{B} = g^{\tilde{r}} = (\tilde{Z}_2 g_5^{-\tilde{t}})^{1/\tilde{h}} = (\tilde{Z}_2 (\tilde{D}_1)^{-\omega_2})^{1/\tilde{h}}$, $g_2^x = \bar{A} \bar{B}^{-\delta - \sum_{i=1}^{\ell} \gamma_i I_i}$. Then $\mathcal{S}$ computes $g^{x^{\ell+1}} = g_2^x g^{-x\gamma}$.

Therefore, the OrcYW Assumption is in the category of *one-more* assumptions, akin to the *one-more RSA Assumption* and the *one-more DL Assumption* [7]. It is more akin to the latter than the former. The state-of-the-art attack on the parallel one-more DL assumption is Schnorr's ROS attack [81] and Wagner's generalized birthday (GB) attack [89]. Below, we examine the plausibility of our OrcYW Assumption against attackers motivated by ROS attackers and GB attackers.

Assume $m_1, \cdots, m_q$ are queried to O_{YW} and the outputs are $(D_{1,\tau}, D_{2,\tau}, Z_{1,\tau}, Z_{2,\tau})$, $1 \leq \tau \leq q$. An attack motivated by Schnorr's ROS [81] attack would attempt to construct $\gamma_1, \cdots, \gamma_q$ satisfying

$$\begin{aligned}
\tilde{D}_1 &= \prod_{\tau} D_{1,\tau}^{\gamma_\tau} = g^{\sum \gamma_\tau t_\tau}, \\
\tilde{D}_2 &= \prod_{\tau} D_{2,\tau}^{\gamma_\tau} = Q^{\sum \gamma_\tau t_\tau}, \\
\tilde{h} &= \sum_{\tau} \gamma_\tau h_\tau, \\
\tilde{Z}_1 &= \prod_{\tau} Z_{1,\tau}^{\gamma_\tau} = g_2^{x \sum \gamma_\tau h_\tau} Q^{\sum \gamma_\tau r_\tau h_\tau} g_4^{\sum \gamma_\tau t_\tau}, \\
\tilde{Z}_2 &= \prod_{\tau} Z_{2,\tau}^{\gamma_\tau} = g^{\sum \gamma_\tau r_\tau h_\tau} g_5^{\sum \gamma_\tau t_\tau},
\end{aligned}$$

The crucial relation is the one about a linear dependence of the hash outputs. Schnorr's blind signature [81] also suffers from similar attack, and he relates the security of the blind signature scheme to the ROS problem.

6.5 Efficient HIBSC without Random Oracles

Motivated by [1]'s generic composition of SignCryption from Encrypt and Sign, we present a generic composition of HIBSC from HIBE and HIBS. Its security is argued below. Then we present a concrete instantiation by composing a HIBSC from [15]'s HIBE and our HIBS in Section 6.3. The security of this specific HIBSC is reduced to a combination of the securities of respective components. The result is a provable HIBSC with size $O(\lambda_s)$

bits which is independent of the levels in the HIBSC. Its security is provable without random oracles, albeit in a weaker model concerning assumptions on the attacker's ability to maneuver identities in the oracles.

6.5.1 Generic Composition from HIBE and HIBS

The generic composition of signcryption from a CCA-secure encryption and an ACP-secure signature is proposed by [1]. They show the security of the outcome without insider attacks. They also give the guidelines of *whenever signing include receiver identity in message* and *whenever encrypting include sender identity in plaintext*, and argued the result would be secure against insider attacks. Motivated by their result, we present a generic composition of HIBSC from HIBE and HIBS.

In [1], a secure signcryption can be composed of a secure signature Sig and a secure encryption Enc via the *sign-then-encrypt* paradigm as follows:

$$\sigma = \mathsf{Enc}_R(\mathsf{Sig}_S(m, \mathsf{ID}_R), \mathsf{ID}_S)$$

where S is the sender and R is the recipient. We observe that such composition can be applied to HIBE and HIBS by treating Enc as the HIBE encryption algorithm and Sig as the HIBS signing algorithm. If [1]'s security theorem for multi-user signcryption is valid, and the hierarchical key derivation system does not cause any problems, then we are likely to have security for the composed HIBSC.

Remarks: In [1], their security is actually for generalized CCA (gCCA), which is a slight relaxation of CCA security. For simplicity, we only mention the CCA security here.

6.5.2 Concrete Instantiation

We give a concrete instantiation of HIBSC from our proposed HIBS, the constant size HIBE from [15] and the transformation technique in [17].

Boneh et al. [17] showed that an adaptive CCA-secure ℓ-level hierarchical identity based encryption (HIBE) scheme Π can be constructed from a CPA-secure $\ell + 1$-level HIBE scheme Π' and a strong one-time signature scheme Sig. The intuition behind their construction is that Π' uses the key extraction oracle to simulate the decryption oracle of Π. If Π wants to query the gauntlet identity, he must have to forge a signature of Sig. Boneh et al. further suggest that a secure encapsulation scheme and a secure message authentication code can be used together in order to replace the strong one-time signature scheme.

As a result, we obtain a constant size HIBSC secure without random oracles. We use [17]'s instantiation of encapsulation scheme. The instantiation is given below:

Setup, Der: same as section 6.3. In addition, let $(\mathsf{Mac}, \mathsf{Vfy})$ be a message authentication code. We assume all signers get the secret keys from master key A $g_2^{\alpha_A}$ and all recipients get the secret keys from master key B $g_2^{\alpha_B}$. Therefore we have $g_{1A} = g^{\alpha_A}$ and $g_{1B} = g^{\alpha_B}$. All other public parameters remain the same as section 6.3.

Signcrypt: For a user with identity $\mathsf{ID}_A = (\mathsf{id}_1, \ldots, \mathsf{id}_k)$ and private key SK_{ID_A}, he signcrypts a message M to recipient $I_B = (I_1, \ldots, I_j)$ as follows.

He picks random $t, \chi \in \mathbb{Z}_p$, and computes:

$$\begin{aligned}
C_1 &= g^t,\ \ C_2 = (h_1^{\mathsf{id}_1} \cdots h_k^{\mathsf{id}_k} \cdot g_3)^t,\ I_{j+1} = \mathcal{H}_3(\chi),\ k_1 = \mathcal{H}_4(\chi), \\
h &= \mathcal{H}(C_1, C_2, \mathsf{ID}_A, I_B, \mathsf{id}', M, \mathsf{param}),\ C_3 = {a_0}^h g_4^t,\ C_4 = {a_1}^h g_5^t, \\
C_5 &= \mathcal{H}_2(\hat{e}(g_{1B}, g_2)^t) \oplus \langle M, \mathsf{ID}_A, C_2, C_3, C_4, \chi\rangle,\ C_6 = (h_1^{I_1} \cdots h_j^{I_j} h_{j+1}^{I_{j+1}} \cdot g_3)^t, \\
C_7 &= \mathsf{tag} = \mathsf{Mac}_{k_1}(\mathsf{ID}_A, I_B, I_{j+1}, C_1, \cdots, C_6)
\end{aligned}$$

The ciphertext σ is $(C_1, C_5, C_6, C_7, I_{j+1})$. Generically,

$$C_5 = \mathsf{SKE.Enc}(\mathsf{key} = \hat{e}(g_1, g_2)^t,\ \ \mathsf{ptxt} = \langle M, \mathsf{ID}_A, C_2, C_3, C_4\rangle)$$

We have adopted Boneh et al.'s [17] tag design above.

Unsigncrypt: The recipient I_B with private key $SK_{I_B} = (a_0, a_1, b_{j+1}, \ldots, b_\ell)$ receives a ciphertext σ $(C_1, C_5, C_6, C_7, I_{j+1})$, he computes:

$$\begin{aligned}
W &= \hat{e}(C_1, a_0 b_{j+1}^{I_{j+1}})/\hat{e}(a_1, C_6) \quad \langle M, \mathsf{ID}_A, C_2, C_3, C_4, \chi\rangle = C_5 \oplus \mathcal{H}_2(W) \\
h &= \mathcal{H}(C_1, C_2, \mathsf{ID}_A, I_B, I_{j+1}, M, \mathsf{param})
\end{aligned}$$

Denote $\mathsf{ID}_A = (\mathsf{id}_1, \ldots, \mathsf{id}_k)$. The recipient computes $k_1 = \mathcal{H}_4(\chi)$ and checks if:

$$\begin{aligned}
\hat{e}(g, C_3) \cdot \hat{e}(g_5, C_2) &\stackrel{?}{=} \hat{e}(g_{1A}, g_2)^h \cdot \hat{e}(C_1, g_4) \cdot \hat{e}(C_4, h_1^{\mathsf{id}_1} \cdots h_k^{\mathsf{id}_k} \cdot g_3) \\
1 &\stackrel{?}{=} \mathsf{Vfy}_{k_1}(\mathsf{ID}_A, I_B, I_{j+1}, C_1, \cdots, C_6, C_7) \\
I_{j+1} &\stackrel{?}{=} \mathcal{H}_3(\chi)
\end{aligned}$$

The recipient outputs M if they are all true. Otherwise, he outputs $\perp$.

Security Analysis

Our HIBSC scheme is secure without random oracles. In particular, it can also imply a secure identity based signcryption scheme without random oracles.

Proposition 1 *Our HIBSC scheme is correct, sID-IND-CCA secure and ACP-UF secure assuming the decisional wBDHI* assumption and the Or-cYW assumption holds.*

The correctness is straightforward.

For indistinguishability, combining the sID-IND-CPA proofs in [15], the transformation theorem in [17] and also the composition theorem of signature and encryption in [1], it implies that our HIBSC is sID-IND-CCA secure.

For existential unforgeability, the HIBSC scheme is ACP-UF secure by Theorem 12 and the composition theorem of signature and encryption in [1].

6.6 Chapter Conclusion

We presented the first constant-size HIBS and HIBSC provable without random oracles. In the reductionist security proofs, we either use an interactive intractability assumption, or use the gID models. We also need the sID model for HIBSC security proof. It is an open problem to avoid these models and assumptions. [1]

[1] After the completion of this research, [77] has proposed an IBS in the standard model. However their security proof is not tight. There is a tradeoff between security and efficiency. Their hierarchical version is believed to be the same.

Chapter 7

Conclusion

In this thesis, we have proposed three schemes in identity based cryptography using pairings. They are *blind identity based signcryption*, *identity based group signatures* and *constant-size hierarchical identity based signatures without random oracles.*

Blind identity based signcryption is a new cryptosystem in identity based cryptography. We propose the security notions and models for it. We give an efficient identity based signcryption scheme, and then extend the scheme to be a new blind identity based signcryption scheme.

Identity based group signatures has not been clearly defined in the literature. We point out the deficiency of the existing definitions. We proceed to give a precise definition and security models. Finally we give an instantiation according to these new definition and models.

Existing hierarchical identity based signatures are mostly provably secure in the random oracle model. Besides, the signature size is dependent on the level of hierarchy. We propose a new constant-size hierarchical identity based signature scheme without using random oracles in the security proofs. Finally, we extend the scheme to a constant-size hierarchical identity based signcryption without random oracles.

Bibliography

[1] J. H. An, Y. Dodis, and T. Rabin. On the Security of Joint Signature and Encryption. In *Proc. EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 83–107. Springer-Verlag, 2002.

[2] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In *Proc. CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–270. Springer-Verlag, 2000.

[3] B. Barak. How to Go Beyond the Black-Box Simulation Barrier. In *FOCS 2001*, pages 106–115. IEEE Computer Soceity, 2001.

[4] B. Barak, Y. Lindell, and S. P. Vadhan. Lower Bounds for Non-Black-Box Zero Knowledge. In *FOCS 2003*, pages 384–393. IEEE Computer Soceity, 2003.

[5] M. Bellare, A. Boldyreva, and A. Palacio. An Uninstantiable Random-Oracle-Model Scheme for a Hybrid-Encryption Problem. In *Proc. EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 171–188. Springer-Verlag, 2004.

[6] M. Bellare, C. Namprempre, and G. Neven. Security Proofs for Identity-Based Identification and Signature Schemes. In *Proc. EU-*

ROCRYPT 2004, volume 3027 of *Lecture Notes in Computer Science*, pages 268–286. Springer-Verlag, 2004.

[7] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme. *Journal of Cryptology*, pages 185–215, 2003.

[8] M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73. ACM Press, 1993.

[9] M. Bellare, H. Shi, and C. Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. In *Proc. CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 136–153. Springer-Verlag, 2005.

[10] T. Beth. Efficient Zero-Knowledged Identification Scheme for Smart Cards. In *Proc. EUROCRYPT 88*, volume 330 of *Lecture Notes in Computer Science*, pages 77–86. Springer-Verlag, 1988.

[11] A. Boldyreva. Efficient Threshold Signature, Multisignature and Blind Signature Schemes Based on the Gap Diffie-Hellman Group Signature Scheme. In *Proc. PKC'03*, volume 567 of *Lecture Notes in Computer Science*, pages 31–46. Springer-Verlag, 2003.

[12] D. Boneh and X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *Proc. EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer-Verlag, 2004.

[13] D. Boneh and X. Boyen. Secure Identity Based Encryption Without Random Oracles. In *Proc. CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459. Springer-Verlag, 2004.

[14] D. Boneh and X. Boyen. Short Signatures Without Random Oracles. In *Proc. EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer-Verlag, 2004.

[15] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical Identity Based Encryption with Constant Size Ciphertext. In *Proc. EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer-Verlag, 2005.

[16] D. Boneh, X. Boyen, and H. Shacham. Short Group Signatures. In *Proc. CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer-Verlag, 2004.

[17] D. Boneh, R. Canatti, S. Halevi, and J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. `http://crypto.stanford.edu/~dabo/abstracts/ccaibejour.html`, 2005.

[18] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. In *Proc. CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer-Verlag, 2001.

[19] D. Boneh and J. Katz. Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption. In *Proc. CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 87–103. Springer-Verlag, 2005.

[20] D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. In *Proc. ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532. Springer-Verlag, 2001.

[21] D. Boneh, I. Mironov, and V. Shoup. A Secure Signature Scheme from Bilinear Maps. In *Proc. CT-RSA 2003*, volume 2612 of *Lecture Notes in Computer Science*, pages 98–110. Springer-Verlag, 2003.

[22] X. Boyen. Multipurpose Identity-Based Signcryption (A Swiss Army Knife for Identity-Based Cryptography). In *Proc. CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 383–399. Springer-Verlag, 2003.

[23] X. Boyen and B. Waters. Compact Group Signatures Without Random Oracles. To appear in EUROCRYPT 2006, 2005. Also available at `http://eprint.iacr.org/2005/381`.

[24] E. F. Brickell, J. Camenisch, and L. Chen. Direct Anonymous Attestation. In *ACM Conference on Computer and Communications Security, CCS 2004*, pages 132–145. ACM Press, 2004.

[25] D. Brown. *The Da Vinci Code*. Doubleday, Mar 2003.

[26] J. Camenisch and I. Damgård. Verifiable Encryption, Group Encryption, and Their Applications to Separable Group Signatures and Signature Sharing Schemes. In *Proc. ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 331–345. Springer-Verlag, 2000.

[27] J. Camenisch and A. Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. In *Proc. CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*. Springer-Verlag, 2004.

[28] J. Camenisch and M. Stadler. Efficient Group Signature Schemes for Large Groups (Extended Abstract). In *Proc. CRYPTO 97*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424. Springer-Verlag, 1997.

[29] R. Canetti, O. Goldreich, and S. Halevi. The Random Oracle Methodology, Revisited. In *Proc. 13th ACM Symp. on Theory of Computing*, pages 209–128. ACM Press, 1998.

[30] R. Canetti, S. Halevi, and J. Katz. A Forward-Secure Public-Key Encryption Scheme. In *Proc. EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 255–271. Springer-Verlag, 2003.

[31] R. Canetti, S. Halevi, and J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *Proc. EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222. Springer, 2004.

[32] C. Castelluccia. How to Convert any ID-based Signature Schemes into a Group Signature Scheme. Cryptology ePrint Archive, Report 2002/116, 2002. `http://eprint.iacr.org/`.

[33] J. Cha and J. Cheon. An Identity-Based Signature from Gap Diffie-Hellman Groups. In *Proc. PKC'2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 18–30. Springer-Verlag, 2003.

[34] D. Chaum. Blind Signatures for Untraceable Payments. In *Proc. CRYPTO 82*, pages 199–203. Plenum Press, 1982.

[35] D. Chaum and E. van Heyst. Group Signatures. In *Proc. EUROCRYPT 91*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer-Verlag, 1991.

[36] L. Chen and J. Malone-Lee. Improved Identity-Based Signcryption. In *Proc. PKC 2005*, volume 3386 of *Lecture Notes in Computer Science*, pages 362–379. Springer-Verlag, 2005.

[37] X. Chen, F. Zhang, and K. Kim. A New ID-Based Group Signature Scheme from Bilinear Pairings. In *WISA2003*, pages 585–592, 2003.

[38] S. S. Chow, J. K. Liu, V. K. Wei, and T. H. Yuen. Ring Signatures without Random Oracles. In *ASIACCS 06*, pages 297–302. ACM Press, 2006.

[39] S. S. Chow, S. Yiu, L. Hui, and K. Chow. Efficient Forward and Provably Secure ID-Based Signcryption Scheme with Public Verifiability and Public Ciphertext Authenticity. In *Proc. ICISC 2003*, volume 2971 of *Lecture Notes in Computer Science*, pages 352–369. Springer-Verlag, 2003.

[40] S. S. Chow, T. H. Yuen, L. C. Hui, and S. Yiu. Signcryption in Hierarchical Identity Based Cryptosystem. In *20th IFIP International Information Security Conference (SEC 2005)*, pages 443–457. Springer-Verlag, 2005.

[41] R. Cramer and V. Shoup. Signature Schemes Based on the Strong RSA Assumption. In *6th ACM Conference on Computer and Communications Security*, pages 46–51. ACM Press, 1999.

[42] Y. Desmedt and J.-J. Quisquater. Public-key Systems Based on the Difficulty of Tampering. In *Proc. CRYPTO 86*, volume 435 of *Lecture Notes in Computer Science*, pages 111–117. Springer-Verlag, 1986.

[43] Y. Dodis, M. J. Freedman, S. Jarecki, and S. Walfish. Versatile Padding Schemes for Joint Signature and Encryption. In *ACM Conference on*

Computer and Communications Security 2004, pages 344–353. ACM Press, 2004.

[44] Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. Anonymous Identification in Ad Hoc Groups. In *Proc. EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 609–626. Springer-Verlag, 2004.

[45] A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification And Signature Problems. In *Proc. CRYPTO 86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer-Verlag, 1986.

[46] M. Fischlin. The Cramer-Shoup Strong-RSA Signature Scheme Revisited. In *PKC 2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 116–129. Springer-Verlag, 2003.

[47] M. Fischlin and R. Fischlin. The Representation Problem Based on Factoring. In *Proc. CT-TSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 96–113. Springer-Verlag, 2002.

[48] G. Frey, M. Müller, and H. Rück. The Tate Pairing and the Discrete Logarithm Applied to Elliptic Curve Cryptosystems, 1999. IEEE Transactions on Information Theory, 45(5):1717–1719.

[49] R. Gennaro, S. Halevi, and T. Rabin. Secure Hash-and-Sign Signatures Without the Random Oracle. In *Proc. EUROCRYPT 99*, volume 1592 of *Lecture Notes in Computer Science*, pages 123–139. Springer-Verlag, 1999.

[50] C. Gentry and A. Silverberg. Hierarchical ID-Based Cryptography. In *Proc. ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer-Verlag, 2002.

[51] M. Girault. An Identity-based Identification Scheme Based on Discrete Logarithms Modulo a Composite Number. In *Proc. EUROCRYPT 90*, volume 473 of *Lecture Notes in Computer Science*, pages 481–486. Springer-Verlag, 1990.

[52] O. Goldreich. *Foundations of Cryptography*, volume 1 and 2. Cambridge Univesity Press, 2001 and 2005.

[53] S. Goldwasser and Y. T. Kalai. On the (In)security of the Fiat-Shamir Paradigm. In *FOCS 2003*, pages 102–113. IEEE Computer Soceity, 2003.

[54] L. C. Guillou and J.-J. Quisquater. A "Paradoxical" Identity-Based Signature Scheme Resulting From Zero-Knowledge. In *Proc. CRYPTO 88*, volume 403 of *Lecture Notes in Computer Science*, pages 216–231. Springer-Verlag, 1989.

[55] S.-H. Heng and K. Kurosawa. k-Resilient Identity-Based Encryption in the Standard Model. In *Proc. CT-RSA 2004*, volume 2964 of *Lecture Notes in Computer Science*, pages 67–80. Springer-Verlag, 2004.

[56] F. Hess. Efficient Identity Based Signature Schemes Based on Pairings. In *Selected Area in Cryptography, SAC2002*, volume 2595 of *Lecture Notes in Computer Science*, pages 310–324. Springer-Verlag, 2003.

[57] F. Hess, N. Smart, and F. Vercauteren. The Eta Pairing Revisited. Cryptology ePrint Archive, Report 2006/110, 2006. `http://eprint.iacr.org/`.

[58] J. Horwitz and B. Lynn. Toward Hierarchical Identity-Based Encryption. In *Proc. EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 466–481. Springer-Verlag, 2002.

[59] A. Joux. A One Round Protocol for Tripartite Diffie-Hellman. In *Algorithm Number Theory Symposium - ANTS IV*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394. Springer-Verlag, 2000.

[60] M. Joye. On the Difficulty of Coalition-Resistance in Group Signature Schemes (II). *Technique Report TR-99-6B*, Tamkang LCIS, 1999.

[61] M. Joye, S. Kim, and N. Lee. Cryptanalysis of Two Group Signature Schemes. In *Information Security (ISW'99)*, volume 1729 of *Lecture Notes in Computer Science*, pages 271–275. Springer-Verlag, 1999.

[62] Laozi. *Tao Te Ching, ch. 33.* about BC 300.

[63] B. Libert and J.-J. Quisquater. New Identity Based Signcryption Schemes from Pairings. IEEE Information Theory Workshop 2003, pages 155-158, 2003.

[64] B. Libert and J.-J. Quisquater. The Exact Security of an Identity Based Signature and its Applications. Cryptology ePrint Archive, Report 2004/102, 2004. `http://eprint.iacr.org/`.

[65] A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf. Pseudonym Systems. In *Selected Areas in Cryptography, SAC '99)*, volume 1758 of *Lecture Notes in Computer Science*, pages 184–199. Springer-Verlag, 1999.

[66] J. Malone-Lee. Identity-Based Signcryption. Cryptology ePrint Archive, Report 2002/098, 2002. `http://eprint.iacr.org/`.

[67] W. Mao. *Modern Cryptography: Theory and Practice.* Prentice-Hall, 2004.

[68] W. Mao and C. Lim. Cryptanalysis in Prime Order Subgroups of Z_n. In *Proc. ASIACRYPT 98*, volume 1514 of *Lecture Notes in Computer Science*, pages 214–226. Springer-Verlag, 1998.

[69] U. Maurer and Y. Yacobi. Non-Interactive Public-key Cryptography. In *Proc. EUROCRYPT 91*, volume 547 of *Lecture Notes in Computer Science*, pages 498–507. Springer-Verlag, 1991.

[70] N. McCullagh and P. S. L. M. Barreto. Efficient and Forward-Secure Identity-Based Signcryption. Cryptology ePrint Archive, Report 2004/117, 2004. `http://eprint.iacr.org/`.

[71] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1996.

[72] D. Nalla and K. Reddy. Signcryption Scheme for Identity-Based Cryptosystems. Cryptology ePrint Archive, Report 2003/066, 2003. `http://eprint.iacr.org/`.

[73] V. Nechaev. Complexity of a Determinate Algorithm for the Discrete Logarithm. *Mathematical Notes 55*, pages 165–172, 1994.

[74] K. Ohta and T. Okamoto. A Modification of the Fiat-Shamir Scheme. In *Proc. CRYPTO 88*, volume 403 of *Lecture Notes in Computer Science*, pages 232–243. Springer-Verlag, 1990.

[75] T. Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In *Proc. CRYPTO 92*, volume 740

of *Lecture Notes in Computer Science*, pages 31–53. Springer-Verlag, 1993.

[76] S. Park, S. Kim, and D. Won. ID-Based Group Signature. In *Electronics Letters, 1997, 33(15)*, pages 1616–1617, 1997.

[77] K. G. Paterson and J. C. N. Schuldt. Efficient Identity-based Signatures Secure in the Standard Model. To appear in ACISP 2006. Cryptology ePrint Archive, Report 2006/080, 2006. `http://eprint.iacr.org/`.

[78] R. L. Rivest, A. Shamir, and Y. Tauman. How to Leak a Secret. In *Proc. ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer-Verlag, 2001.

[79] A. Sahai and B. Waters. Fuzzy Identity-Based Encryption. In *Proc. EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer-Verlag, 2005.

[80] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems Based on Pairing. In *Proc. of the 2000 Symposium on Cryptography and Information Security*, pages 26–28, 2000.

[81] C.-P. Schnorr. Security of Blind Discrete Log Signatures against Interactive Attacks. In *ICICS 2001*, volume 2229 of *Lecture Notes in Computer Science*, pages 1–12. Springer-Verlag, 2001.

[82] M. Scott, N. Costigan, and W. Abdulwahab. Implementing Cryptographic Pairings on Smartcards. Cryptology ePrint Archive, Report 2006/144, 2006. `http://eprint.iacr.org/`.

[83] A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *Proc. CRYPTO 84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer-Verlag, 1984.

[84] V. Shoup. Lower Bounds for Discrete Logarithms and Related Problems. In *Proc. EUROCRYPT 97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer-Verlag, 1997.

[85] M. Sipser. *Introduction to the Theory of Computation.* PWS Publishing, 1997.

[86] H. Tanaka. A Realization Scheme for the Identity-Based Cryptosystem. In *Proc. CRYPTO 87*, volume 293 of *Lecture Notes in Computer Science*, pages 341–349. Springer-Verlag, 1987.

[87] Y. Tseng and J. Jan. A Novel ID-Based Group Signature. In *International computer symposium, workshop on cryptology and information security*, pages 159–164, 1998.

[88] S. Tsujii and T. Itoh. An ID-based Cryptosystem based on the Discrete Logarithm Problem. In *IEEE Journal on Selected Areas in Communication, vol.7, no.4*, pages 467–473, 1989.

[89] D. Wagner. A Generalized Birthday Problem. In *Proc. CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 288–303. Springer-Verlag, 2002.

[90] B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In *Proc. EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer-Verlag, 2005.

[91] V. K. Wei. Tight Reductions among Strong Diffie-Hellman Assumptions. Cryptology ePrint Archive, Report 2005/057, 2005. `http://eprint.iacr.org/`.

[92] V. K. Wei and T. H. Yuen. More Short Signatures without Random Oracles. Cryptology ePrint Archive, Report 2005/463, 2005. `http://eprint.iacr.org/`.

[93] V. K. Wei, T. H. Yuen, and F. Zhang. Group Signature Where Group Manager, Members and Open Authority Are Identity-Based. In *Proc. ACISP 2005*, volume 3574 of *Lecture Notes in Computer Science*, pages 468–480. Springer-Verlag, 2005.

[94] Wikipedia. Self (philosophy) — Wikipedia, The Free Encyclopedia, 2006. [Online; accessed 12-May-2006, `http://en.wikipedia.org/w/index.php?title=Self_%28philosophy%29&oldid=51879956`].

[95] T. H. Yuen and V. K. Wei. Constant-Size Hierarchical Identity-Based Signature/Signcryption without Random Oracles. Cryptology ePrint Archive, Report 2005/412, 2005. `http://eprint.iacr.org/`.

[96] T. H. Yuen and V. K. Wei. Fast and Proven Secure Blind Identity-Based Signcryption from Pairings. In *Proc. CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 305–322. Springer-Verlag, 2005.

[97] F. Zhang, X. Chen, W. Susilo, and Y. Mu. A New Short Signature Scheme Without Random Oracles from Bilinear Pairings. Cryptology ePrint Archive, Report 2005/386, 2005. `http://eprint.iacr.org/`.

[98] F. Zhang and K. Kim. ID-Based Blind Signature and Ring Signature from Pairings. In *Proc. ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 533–547. Springer-Verlag, 2002.

[99] F. Zhang and K. Kim. Efficient ID-Based Blind Signature and Proxy Signature from Bilinear Pairings. In *Proc. ACISP'03*, volume 2727 of *Lecture Notes in Computer Science*, pages 312–323. Springer-Verlag, 2003.

[100] F. Zhang, R. Safavi-Naini, and W. Susilo. Efficient Verifiably Encrypted Signature and Partially Blind Signature from Bilinear Pairings. In *Proc. INDOCRYPT03*, volume 2904 of *Lecture Notes in Computer Science*, pages 191–204. Springer-Verlag, 2003.

[101] F. Zhang, R. Safavi-Naini, and W. Susilo. An Efficient Signature Scheme from Bilinear Pairings and Its Applications. In *Proc. PKC'2004*, volume 2947 of *Lecture Notes in Computer Science*, pages 277–290. Springer-Verlag, 2004.

[102] Y. Zheng. Digital Signcryption or How to Achieve Cost(Signature & Encryption) $<<$ Cost(Signature) + Cost(Encryption). In *Proc. CRYPTO 97*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer-Verlag, 1997.