Identity-based Cryptography from Paillier Cryptosystem

AU Man Ho Allen

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

本論文研究 Paillier 密碼系統。大部分密碼系統根據的活板門函數爲 RSA 或離算對數問題(discrete logarithm)。Paillier 研究別的活板門函數——Composite degree residuosity class。Paillier 密碼系統的安全性以 RSA 問題爲依歸，而其特性對密碼學十分有用。

身份碼密碼驗證(Identity-based identification)系統讓用戶識別被驗證者的身份。根據 Paillier 系統，本論文提出數個身份碼密碼驗證程序。我們提出的程序可用作電子簽署系統。同時，我們把一個現行的身份碼密碼系統融入我們的系統中。

我們並爲系統的安全性提供理論上的證明。

Abstract of thesis entitled:

Identity-Based Cryptography from Paillier Cryptosystem

Submitted by AU Man Ho Allen

for the degree of Master of Philosophy

at The Chinese University of Hong Kong in June 2005

Majority of cryptographic systems relies on one of the two trapdoor mechanism, namely, RSA and discrete logarithm. Paillier studied cryptosystem based on other trapdoor mechanism, the composite degree residuosity class, and proposed the Paillier cryptosystem.

This dissertation studies the Paillier cryptosystem. Although it turns out that Paillier cryptosystem relies on the difficulty of computing the RSA problem, the trapdoor mechansim from Paillier is useful for many applications.

Identity-based identification schemes allows users to prove their identities to verifiers. Several efficient realizations of the concept, based on Paillier Cryptosystem, are being proposed. Furthermore, our constructions can be turned into identity-based signature schemes easily using the Fiat-Shamir heuristic. We also reformat the identity-based encryption scheme from Cocks to make it compatible with our setting.

We provide evidence that our constructions are secure by presenting reduction proofs in the random oracle model. Security of our constructions depends on well-studied hard problems.

Acknowledgement

I would like to thank my supervisor Prof. Victor Wei, who gave me the opportunity to work in the area of cryptography. I would also like to thank Prof. Duncan Wong and Joseph Liu for their support in my studies.

It has been nice to share the office with my colleges. Room 726 has been the place we spent our lives together for two years. Special thanks to Patrick Tsang, who discuss with me a lot in cryptography and those discussions are fruitful. And those free meals he offered are fantastic. I am very grateful to Sebastian for being helpful. I thank the not-yet-mentioned members of Information Security Laboratory, Patrick Chan, Robert Leung, Tony Chan, Karyin and Yuen. We have had a great atmosphere in room 726.

Many thanks to everybody who have helped me, being kind to me and being friendly to me. Friends are important for me in my life.

Last but not least, I would like to express my gratitude to my parents for everything, everything and everything. I would also like to thank Cindy for always being on my side.

This work is dedicated to Sandra and David.

Contents

Chapter 1

Introduction

Diffie and Hellman started the revolution in cryptography with their classic paper "New Directions in Cryptography"[14] in 1976. They invented the concept of public key cryptography and make secret communication possible over insecure channel without a prior exchange of a secret key.

Consider the situation when Alice wishes to communicate with Bob over an insecure channel. In public key cryptography, Alice request Bob to send his public key e to Alice first. She then encrypts the message using e. No one other than Bob can decrypt the message because only he know the private key d. In this way, they can communicate secretly over any public channel.

However, opponent Oscar can still defeat the system by impersonating Bob and send his own public key e' to Alice when she request for Bob's public key. He can then intercept and decrypt the message Alice encrypted using e'. Therefore, it is necessary that Alice must be convinced that she is encrypting under the legitimate public key of Bob. The use of digital certificate is one solution to the problem. Instead of sending Alice Bob's public key, Bob can send his digital certificate that contains his public key. The solution is, however, somehow tedious.

In 1984, Shamir [39] proposed the idea of using the identity of the recipient as public key directly. This is known as identity-based cryptography. Back to our example, when Alice wishes to communicate with Bob, she simply encrypt the message using the bit string "Bob" as public key and thus eliminate the request of public key or digital certificate.

On the other hand, the asymmetry of key also make it possible for the development of digital signature. Here, the private key is used to sign a message and the public key is used to verify the signature. A closely related concept is identification protocol for which the owner of a public key shows the verifier that he is the legitimate owner by proving that he knows the secret key correspond to the public key.

Public key cryptography has been a very active research area in the academia. Many realizations of encryption scheme and digital signature scheme were proposed. Paillier encryption and signature scheme [32] is one of which being proposed. Based on these primitives, many more complex systems are being devised.

This dissertation is about Identity-based identification scheme based on Paillier cryptosystem. The rest of this thesis is organized as follow. Chapter 2 provides the mathematical and cryptographical background. This includes number theory, Algeria and complexity theory. A brief introduction to public key cryptography is also given.

In Chapter 3 we talk about the Paillier cryptosystem for which our results are based on. We talk about the background of Paillier cryptosystem and outline what it is. Then we discuss several encryption schemes related to Paillier cryptosystem.

Chapter 4 is about Identity-based cryptography. We review Identity-based encryption scheme, signature scheme and identification scheme. Cocks' identity-based encryption scheme[11] is also discussed here.

In Chapter 5 we presented our constructions of identity-based identification scheme from Paillier cryptosystem. We also reformat Cocks' identity-based encryption scheme in Paillier setting.

We concluded in Chapter 6 by giving certain possible future research directions.

□ **End of chapter.**

Chapter 2

Preliminaries

Summary

This chapter introduces topics of complexity theory, number theory and cryptography that will be used in subsequent chapters. Readers interested in the theory of cryptography will find Oded Goldreich's book "Foundations of Cryptography" [18] and Wenbo Mao's book "Modern Cryptography: Theory and Practice" [25] helpful.

2.1 Complexity Theory

Let $\mathcal{A}$ be an algorithm. By $\mathcal{A}(\cdot)$ (resp. $\mathcal{A}(\cdot, \ldots, \cdot)$) we denote that $\mathcal{A}$ has one input (resp. several inputs). $y \leftarrow \mathcal{A}(x)$ denotes that y was obtained from algorithm $\mathcal{A}$ on input x.

In complexity theory, problems are classified by the most efficient algorithm that solve them. Efficiency of an algorithm is measured by the resources required to solve the problem. Time complexity (resp. space complexity) of an algorithm refers to the number of primitive steps (resp. memory) required to solve the problem.

Standard asymptotic notation is used to compare running time of algorithms. By $f(n) = \mathcal{O}(g(n))$ we denote that there exists some positive constants c, n_0 such that for all $n \geq n_0$, $0 \leq f(n) \leq cg(n)$. That is, f is bounded asymptotically by g. If $g(n) = \mathcal{O}(f(n)$ holds, then $f(n) = \Omega(g(n))$. Furthermore, if $f(n) = \mathcal{O}(g(n))$ and $g(n) = \mathcal{O}(f(n))$, then we write $f(n) = \Theta(g(n))$. On the other hand, $f(n) = o(g(n))$ means that the upper-bound is not asymptotically tight. That is, for any positive constant c, there exists an integer n_0 such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n_0$.

Let $\mathcal{A}$ be an algorithm with running time of $\mathcal{A}$ being $\mathcal{O}(\exp(c+o(1)n^\alpha(\ln n)^{1-\alpha}))$ for some positive constant c, α, satisfying $0 < \alpha < 1$ with respect to input size n. We say that $\mathcal{A}$ is polynomial-time if $\alpha = 0$, exponential-time if $\alpha = 1$ and sub-exponential time otherwise.

2.2 Algebra and Number Theory

Number theory plays an important role in public key cryptography. We review some of the basic facts that shall be used in subsequent sections.

2.2.1 Groups

A group is a non-empty set S together with a binary operation $*$ that maps $S \times S$ to S satisfying the following properties.

- Associative: $(a * b) * c = a * (b * c)\ \forall a, b, c \in S$
- Existence of Identity: $\exists u \in S$ s.t. $\forall a \in S$, $a * u = u * a = a$
- Existence of Inverse: $\forall a \in S$, $\exists b \in S$ s.t. $a * b = u \in S$. b is called the inverse of a

In addition, if $a * b = b * a \forall a, b \in S$, then it is called a commutative (or abelian) group. If the binary operation is called addition (denoted by $+$), the identity element is denoted by 0 and inverse element of a is denoted by $-a$. On the other hand, if the operation is multiplication, the inverse of a is denoted by $1/a$ or a^{-1}. We use the notation a^n for element a multiplying itself n times and a^{-n} to denote element a^{-1} multiply itself by n times.

Let G be a group. $|G|$ denotes the number of elements G. G is finite if $|G|$ is finite and G is cyclic if $\exists g \in G$ s.t. $\forall a \in G$ $\exists x \in \mathbb{Z}$ s.t. $a = g^x$. g is called generator of G and we can write $\langle g \rangle = G$. The order of an element a, denoted by ord(a), is the smallest positive integer n such that $a^n = 1$. A group H is said to be a subgroup of another group G, denoted by $H \subseteq G$, if H and G shares the same binary operation and $\forall a \in H, a \in G$.

2.2.2 Additive Group $\mathbb{Z}_n$ and Multiplicative Group $\mathbb{Z}_n^*$

One important group in cryptography is the set of integers modulo n together with addition modulo n. This group, denoted by $\mathbb{Z}_n$, is abelian. Another important group $\mathbb{Z}_n^*$ is formed by the set of positive integers smaller than n and relatively prime to n with multiplication modulo n. It is obvious that $|\mathbb{Z}_n| = n$ and $|\mathbb{Z}_n^*| = \phi(n)$ where the Euler totient function $\phi(n)$ is defined as follow.

Definition 2.1. *The Euler totient function $\phi(n)$ for any positive integer n is $\phi(n) = |\{a | 1 \leq a < n, gcd(a, n) = 1\}|$.*

For $n = \prod(p_i)^{\alpha_i}$, where p_i are the prime factors of n, $\phi(n)$ can be computed by

$$\phi(n) = n \prod(1 - 1/p_i)$$

We have the following theorems regarding $\phi(n)$.

Theorem 2.2 (Euler's Totient Theorem).

$$a^{\phi(n)} = 1 \bmod n$$

for all a relatively prime to n.

In particular, if n is a prime number, we have the Fermat's Little Theorem.

Theorem 2.3 (Fermat's Little Theorem).

$$a^{n-1} = 1 \bmod n$$

for all $n \nmid a$ where n is prime.

2.2.3 The Integer Factorization Problem

The security of many cryptosystems, such as RSA[37], Rabin[35], to name a few, relies on the hardness of the integer factorization problem. We first describe when we consider a problem to be hard in an rather informal manner in the following definition. For a more formal treatment, see [25].

Definition 2.4. *A problem is said to be easy when there exists an algorithm that solves the problem with running time that is polynomial in size of the input. A problem is hard when no such algorithm exists.*

Definition 2.5 (Integer Factorization Problem). *Given a positive integer n, find its prime factorization. That is, write $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$ where the p_i are pairwise distinct primes and $\alpha_i \geq 1$*

Some algorithms are tailored to perform better for n of special format. These algorithms, including trail division, Pollard's rho algorithm, Pollard's p-1 and the elliptic curve algorithm, are known as special-purpose factoring algorithm. In contrast, the running time of the general-purpose factoring algorithm depends only on the size of n. Examples of these types of algorithms includes quadratic sieve and general number field sieve.

If a large prime n is the product of two primes which are roughly of the same size, no algorithms are known that can factor in polynomial time. However, sub-exponential time algorithm exists. For example, the number field sieve algorithm[24] has a time complexity of $\mathcal{O}(\exp(1.92 + o(1)(\ln n)^{1/3}(\ln \ln n)^{2/3}))$.

Definition 2.6 (Computing Square Roots Problem). *Let n be a composite number. Given y, find x s.t. $x^2 = y \bmod n$, providing that such x exists.*

The integer factorization problem is equivalent to the problem of computing square root. That is, suppose we have polynomial-time algorithm which can solve the integer factorization problem, we can use it to construct an algorithm which can solve the computing square roots problem and vice versa. In fact, the Rabin public key encryption schemes uses this computational equivalence to achieve the first "provably secure" encryption scheme.

2.2.4 Quadratic Residuosity Problem

Definition 2.7 (Quadratic Residue). *An element $a \in \mathbb{Z}_n^*$ is a quadratic residue modulo n if $\exists x$ such that $x^2 = a \bmod n$. If there exist no such $x \in \mathbb{Z}_n$, a is called a quadratic non-residue. The set of all quadratic residues and the set of all non-residues are denoted by QR_n and QNR_n respectively.*

We uses the Legendre symbol to keep track of whether or not an integer is a quadratic residue modulo a prime number.

Definition 2.8 (Legendre Symbol). *Let p be an odd prime number and a an integer. The legendre symbol, denoted by $(\frac{a}{p})$, is defined to be 0 if $p|a$, 1 if $a \in QR_n$ and -1 if $a \in QNR_n$ respectively.*

We can generalize Legendre symbol for integer n which may not be odd prime as follow.

Definition 2.9 (Jacobi Symbol). *Let n be an integer greater* *than 3 with prime factorization $p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$, and a be an integer. The Jacobi symbol, denoted by $(\frac{a}{n})$, is defined as follow:*

$$(\frac{a}{n}) = (\frac{a}{p_1})^{\alpha_1} (\frac{a}{p_2})^{\alpha_2} \cdots (\frac{a}{p_k})^{\alpha_k}$$

It is worth noting that $(\frac{a}{n}) = 1$ does not imply a is a quadratic residue modulo n. $(\frac{a}{n})$ can be computed efficiently[11] without factorization of n. We define $J_n = \{a \in \mathbb{Z}_n^* | (\frac{a}{n} = 1)\}$

We are now ready to define the quadric residuosity problem (QRP) which is to decide if an integer is a quadratic residue modulo n. The security of Goldwasser-Micali probabilistic public-key encryption scheme[19] relies on this problem.

Definition 2.10 (QRP). *Given an odd positive composite integer n and $a \in J_n$, decide whether or not a is a quadratic residue modulo n.*

It is obvious that if we can solve the integer factorization problem, QRP can be solved efficiently. On the other hand, no algorithm, other than random guessing, is known to solve QRP. If $n = pq$, then the probability of guessing correctly is $1/2$. It is believed that QRP is as hard as factorization[27], although no proof of this is known.

2.2.5 Computing e-th Roots (The RSA Problem)

The hardness of the RSA problem is the basis of the RSA[37] encryption and signature scheme and many other schemes.

Definition 2.11. *Given $n = pq$, where p and q are odd primes, and e such that gcd(e,$\phi(n)$)=1, and an integer c, find an integer m such that $m^e = c \bmod n$. n and e are sometimes called modulus and exponent respectively.*

If integer factorization is easy, then so is the RSA problem. Whether the converse is also true is not known. We shall denote the RSA problem with modulus n and exponent e by RSA[n,e].

2.2.6 Discrete Logarithm and Related Problems

The hardness of discrete logarithm problem is the basis of many cryptosystems.

Definition 2.12. *Let G be a finite cyclic group of order n and $g \in G$ be a generator of G. The discrete logarithm problem (DLP) is define as follow. Given an element $y \in G$, find the integer x, $0 \leq x \leq |G| - 1$, such that $y = g^x$ holds. x is denoted by $\log_g(y)$.*

The generalized discrete logarithm problem is that given a finite group G (not necessarily cyclic), two elements y,h in G, find x such that $y = h^x$ provided such x exists. Just as the case for integer factorization, we shall briefly talk about algorithm that solves DLP. These algorithms can be categorized into the following three categroies.

- Generic Algorithms. The algorithms in this category do not use the properties of the underlying group besides multiplication, inversion and unique encoding of the group elements. Examples include Shanks' Baby-Step Giant-Step method[22], Pollard's rho method[34]. A result of Shoup[40] stated that any generic methods takes at least $O(\sqrt{n})$ operations to solve DLP, where n is the order of the group. Therefore, generic algorithms must be exponential in the size of the input.

- Algorithms which work in arbitrary groups but are especially efficient if the order of the group has only small prime factors. An Example is the Pohling-Hellman algorithm[33].

- Special algorithms that exploit the representation of the group elements. The algorithms in this category work only in the group they were designed for. An example is the Number Field Sieve[41] for the group $\mathbb{Z}_p^*$, where p is prime. Running time of Number Field Sieve is

$$O(\exp((1.92 + o(1))(\ln p)^{1/3}(\ln\ln p)^{2/3})).$$

We would like to point out that hardness of DLP depends strongly on the representation of the elements of the group. Groups on which no attacks other than generic ones are suitable for the design of DL-based cryptographic protocols.

Closely related to the discrete logarithm problem is the computational Diffie-Hellman problem (CDH).

Definition 2.13 (Computational Diffie-Hellman Problem). *Given a finite cyclic group G, a generator g, two elements g^a, g^b, find g^{ab}.*

Obviously, CDH is no harder than DLP. For some groups, CDH and DLP are shown to be computationally equivalent[26].

Besides the computational Diffie-Hellman problem, there exists a weaker version called the decision Diffie-Hellman problem (DDH), introduced in [7].

Definition 2.14 (Decision Diffie-Hellman Problem). *Given a finite cyclic group G, a generator g, three elements g^a, g^b, g^c, decide whether $g^c = g^{ab}$.*

It is obvious that DDH is no harder than CDH. For most groups it is not clear whether DDH is easier than CDH. Certain groups with the property that CDH is hard and DDH is easy are called Gap Diffie-Hellman (GDH) groups.

2.3 Public key Cryptography

In public key cryptography, also known as asymmetric cryptography, each user has a key pair consisting of a public key and a secret key such that given the public key, it is hard to derive the secret key. This is in contrast with secret key cryptography, also known as symmetric cryptography or conventional cryptography, in which there is only a single key or the encryption/decryption key pair can be derived from each other easily.

Symmetric key encryption schemes have been known for ages. Commonly used symmetric-key encryptions include Data Encryption Standard (DES), Advanced Encryption Standard (AES), IDEA, etc. They are efficient and secure, provided that the encryption/decryption key is unknown to adversary. However, the problem of symmetric key encryption schemes is that it is difficult to find an efficient way for two parties to exchange the secret key securely.

Public key cryptography was only invented in 1977 by Diffie and Hellman[14]. In public key cryptography, each user U has a key pair (pk, sk) consisting of a public key and a secret key.

Given pk, it is computationally hard to find sk. In an encryption scheme, other parties uses U's public key pk to encrypt message for U. Only U, who know the secret key sk, can decrypt the message. The development of public-key cryptography is considered a revolution in cryptography: while the key for conventional cryptography must be exchanged securely, the public key only need to be exchanged authentically.

Public key cryptography also make it possible to realize the digital counterpart of handwritten signature: digital signature for electronic files.

2.3.1 Encryption

As mentioned before, each user in an a public key encryption scheme possess a key pair. In fact, a public key encryption scheme is a oneway trapdoor function f with trapdoor information t. A oneway trapdoor function is some function that is easy to compute but hard to invert without the trapdoor information. The idea is that f is used as the public key, and t is use as the secret key. Suppose Bob wants to encrypt a message m to Alice with public key f, Bob computes ciphertext $c = f(m)$ and transmit c to Alice. Alice decrypt by computer $m = f^{-1}(c)$ using her trapdoor information t. Only Alice can do so because of the oneway trapdoor property of f.

Formally speaking, an encryption scheme E is a 3-tuple $(\mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$. Keygen takes security parameter λ to output (pk,sk) where pk is a public key and sk is a secret key. We write $(pk, sk) \leftarrow \mathsf{Keygen}(1^\lambda)$. The encryption algorithm Encrypt output a ciphertext c on input message m and public key pk; we write $c \leftarrow \mathsf{Encrypt}_{pk}(m)$. The decryption algorithm Decrypt output message m or reject on input ciphertext c and secret key sk; we write $x \leftarrow \mathsf{Decrypt}_{sk}(c)$, where x can be m or reject. We required that $\forall (pk, sk) \leftarrow \mathsf{Keygen}(1^\lambda)$, $\mathsf{Decrypt}_{sk}(\mathsf{Encrypt}_{pk}(m)) = m$ for all message m. Keygen, Encrypt, Decryptare all polynomial time algorithms.

There are several concepts of security in public-key encryption. The most basic one being one-way secure which means that given a ciphertext, no polynomial time adversary should be able to obtain the plaintext m from the given ciphertext. This security is called OW-CPA. We are going to consider semantic security and chosen ciphertext security here. That latter is sufficiently strong for most applications and is thus and acceptable notion of security for public key encryption schemes. For a detailed description of security notions, refer to [1].

We say that a public key encryption scheme E is semantic secure against chosen plaintext attack if it is hard to find any (partial) information on message m from ciphertext c. This notion is closely related to indistinguishability against chosen plaintext attack (IND-CPA), which is described as follows.

For IND-CPA security, we consider a game between the dealer and an adversary. Suppose the dealer gives an adversary a random public key. The adversary then comes up with two messages. The dealer chooses one of which randomly and encrypted it as a challenge (gauntlet) ciphertext. If the adversary correctly guesses which one, he wins the game. An encryption scheme is said to be IND-CPA secure if no polynomial time adversary can win the game with probability non-negligibly more than a half.

For chosen ciphertext security, we consider a similar game. Only this time, the adversary is allowed to issue a number of decryption queries to the dealer. We say the adversary is given access to the decryption oracle. That is, the adversary present a ciphertext of his choice to the dealer and the dealer responds with the decryption of that ciphertext under the secret key corresponding to the public key given to the adversary. Of course, the adversary is not allowed to query the gauntlet ciphertext. A public key encryption scheme is said to be IND-CCA2 secure if no polynomial time adversary can win the game with probability non-negligibly more than a half. Intuitively, IND-CCA2 security means that even if the adversary has access to the decryptions of a number of his choice, he still cannot learn anything about the plaintext of a given ciphertext.

2.3.2 Digital Signature

Digital signature scheme is the analogue of handwritten signature. Intuitively, a digital signature must be hard to forge and easy for everyone to verify. A digital signature is in essence a bit string that related the message to the signer's public key.

Formally speaking, a digital signature scheme S is a 3-tuple $(\mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify})$. Keygen takes security parameter λ to output (pk,sk) where pk is a public key and sk is a secret key. We write $(pk, sk) \leftarrow \mathsf{Keygen}(1^\lambda)$. The signing algorithm Sign output a signature σ on input message m and secret key sk; we write $\sigma \leftarrow \mathsf{Sign}_{sk}(m)$. The verification algorithm Verify output 0 or 1 on input message m, signature σ and public key pk; we write $x \leftarrow \mathsf{Verify}_{pk}(\sigma, m)$, where x can be 0 or 1. We required that $\forall (pk, sk) \leftarrow \mathsf{Keygen}(1^\lambda)$, $\mathsf{Verify}_{pk}(\mathsf{Sign}_{sk}(m), m) = 1$ for all message m. In addition, it is required that a signature scheme must be unforgeable. This means that is must be infeasible to compute a signature of a message with respect to a public key without knowing the corresponding secret key. Keygen, Sign, Verifyare all polynomial time algorithms.

The acceptable notion of security for digital signature scheme is existential unforgeability against chosen message attack (uf-cma). We consider a game between the dealer and an adversary as follow. The dealer gives an adversary a random public key. The adversary is allowed to issue a number of signing queries to the dealer. We say the adversary is given access to the signing oracle. That is, the adversary present a message of his choice to the dealer and the dealer responds with a valid signature of that message corresponding to the public key given to the adversary. The adversary wins the game if he could deliver a valid signature and message pair under the public key given by the dealer. Of course, the adversary is not allowed to submit message that has been queried to the dealer for signature. A digital signature scheme is said to be uf-cma secure if no polynomial time adversary can win the game with probability non-negligibly. Intuitively, uf-cma security means that even if the adversary has access to the signer for a number of message of his choice, he still cannot forge a new signature that the signer has not signed.

2.3.3 Identification Protocol

An identification protocol allows a prover Peggy to convince a verifier Victor of her identity. Victor is given the public key belongs to Peggy. If someone could prove to Victor that she knows the secret key corresponding to the Peggy's public key, Victor can concluded that this entity must be Peggy.

Informally speaking, an identification protocols (sometimes known as standard identification protocols SI) is a 3-tuple (Keygen, Prover, Verifier). Keygen takes security parameter λ to output (pk,sk) where pk is a public key and sk is a secret key. We write $(pk, sk) \leftarrow \mathsf{Keygen}(1^\lambda)$. (Prover, Verifier) is an interactive protocol for prover Peggy and verifier Victor. The protocol must satisfy three properties.

- Completeness. Peggy, knowing the secret key, must be able to convince Victor for his identity.
- Soundness. Entity not knowing the secret key must not be able to convince Victor that she is Peggy.
- Zero-knowledgeness. Victor should not be able to learn anything about Peggy's secret key.

In this dissertation, we only consider three-move identification protocols, commonly known as canonical. It means that the interactive protocol between (Prover, Verifier) is of the following form.

1. Prover sends a commitment t to Verifier.
2. Verifier returns a challenge c which is randomly chosen from some set.
3. Prover provides a response z.
4. Based on the input (pk, t, c, z), Verifier output Accept or Reject.

Identification protocol should be secure against impersonation. An adversary succeeds in an impersonation attack if it interacts with the verifier in the role of a prover and can convince the verifier to accept. We consider three types of attackers, namely, passive, active and concurrent attacker. We consider the following two-phase game between the dealer and the adversary. In phase I, adversary is given a random public key for impersonation. Adversary is allowed to make some transcript query(for passive attack) or request to act as a (cheating) verifier (for active and concurrent attack). For transcript query, dealer return a complete communication transcript between a prover and verifier. The difference between active and concurrent attack is that in the former case, request for being (cheating) verifier must be sequential. An identification protocol is imp-atk-secure, where atk$\in${pa,aa,ca} if it is secure against impersonation under passive, active or concurrent attack. That is, no polynomial time adversary can win in the above game.

Identification protocol can be used to construct digital signature schemes by the Fiat-Shamir transform[15]. For such constructions, it is often argued that the resulting signature scheme is uf-cma secure if the underlying identification protocol is imp-pa-secure and a secure one-way hash function is used. The resulting signature scheme is said to be secure in the random oracle model [3].

2.3.4 Hash Function

A hash function H is a transformation that takes a variable-size input m and returns a fixed-size string, which is called the hash value h (that is, $h = H(m)$). Usually, it has to be easily computable.

Hash functions employed in cryptography have at least one of the following properties.

- one-way. For a given h, it is difficult to find x such that $H(x) = c$
- weak collision resistant. For a given x, it is hard to find an

$x' \neq x$ s.t. $H(x) = H(x')$

- strong collision resistant. It is hard to find a pair (x,x'), $x \neq x'$, such that $H(x) = H(x')$

In digital signature schemes, hash function can be used to reduce message size. It can also be used to turn interactive proofs of knowledge protocols into digital signature schemes by taking the place of the verifier. Currently MD5 and SHA-1 are most popular choice of hash functions. Recently, collision of MD5 has been found [21]. A Chinese research team also claimed that SHA-1 is vulnerable and they have developed algorithm to find collision for full SHA-1(whose output is 160 bit) with 2^{69} calculations. Their result has not been published yet at the moment. We will not discuss the issue in further detail in this thesis.

□ **End of chapter.**

Chapter 3

Paillier Cryptosystems

Summary

This chapter introduces Paillier Cryptosystem [32]. Several relevant schemes are also outlined. This chapter provides building blocks for the identification schemes described in the next chapters.

3.1 Introduction

Goldwasser and Micali started the work on trapdoor mechanism based on quadratic residuosity [19] in 1984. Their scheme, however, is bandwidth inefficient. Benaloh and Fischer[12] uses higher order resides to improve the bandwidth efficiency but the decryption is inefficient. In 1998, Naccache and Stern[29] proposed a variant of the Benaloh-Fischer scheme with better bandwidth efficiency. Their scheme make use of residuosity of smooth degree in $\mathbb{Z}_{pq}^*$. At the same time, Okamoto and Uchiyama[31] proposed to use residuosity of prime degree p in the group $\mathbb{Z}_{p^2q}^*$. The scheme has similar bandwidth efficiency as Naccache-Stern but with improved decryption efficiency.

In 1999, Paillier[32] brought re-vigored interests to this trapdoor mechanism in the group of $\mathbb{Z}_{p^2q^2}^*$. Since then, it has found uses in verifiable encryption[9] and double trapdoor decryption[8]. Several variants of Paillier's cryptosystem have been proposed recently [10, 17].

3.2 The Paillier Cryptosystem

Let $n = pq$ be an RSA modulus and g an element having order αn with $\alpha \geq 1$ in the multiplicative group $\mathbb{Z}_{n^2}^*$. To encrypt a message $m \in \mathbb{Z}_{n^2}^*$, Paillier proposed the following mechanism.

$$\varepsilon_g : \mathbb{Z}_n \times \mathbb{Z}_n^* \to \mathbb{Z}_{n^2}^*$$

$$(m_1, m_2) \mapsto g^{m_1} {m_2}^n \bmod n^2$$

where $m = m_1 + m_2 N$ and he proved that:

- ε_g is a bijection between $\mathbb{Z}_n \times \mathbb{Z}_n^*$ and $\mathbb{Z}_{n^2}^*$.

- ε_g is a one-way trapdoor permutation equivalent to RSA[n,n]
- the above is one-way if and only if RSA[n,n] is hard.

For any $w \in \mathbb{Z}^*_{n^2}$, there exists unique $(x, y) \in (\mathbb{Z}_n, \mathbb{Z}^*_n)$ such that $w = \varepsilon_g(x, y)$. Paillier called x the class of w relative to g(denoted by $[w]_g$) and informally, computing $[w]_g$ given w and g is called the computational composite residuosity class problem. If $w \in < g >$, computing $[w]_g$ is called partial discrete logarithm problem (PDL). Paillier assume both of them are hard. Note also that inverting ε_g is equivalent to RSA[n,n]. We also have the following definition with regard to class.

Definition 3.1 (Decisional Composite Residuosity Class Assumption (D-Class) [32]). *Given prime product n, and $W \in \mathbb{Z}^*_{n^2}$, $r \in \mathbb{Z}_n$, it is infeasible to decide with probability over random guessing, in polynomial time, if there exists $y \in \mathbb{Z}^*_n$ such that $W = (1 + n)^r y^n (\text{mod} n^2)$.*

Given $c = g^x y^n \bmod n^2$, x, y can be found as follow. Define

$$L(u) = (u - 1)/n$$

Then compute,

$$l = \text{lcm}(p, q)$$

$$x = (L(c^l \bmod n^2)/L(g^l \bmod n^2)) \bmod n$$

$$y = (cg^{-x})^{n^{-1} \bmod l} \bmod n$$

We outline several Paillier-related encryption schemes. Denote (c, m) as (ciphertext, plaintext) pair. Denote r as random number from $\mathbb{Z}_n$.

CATALANO, ET AL.[10]. $c = (1+n)^m y^e \bmod n^2$, where $(e, \lambda(n)) = 1$. Its one-wayness is reducible to RSA[n,e].

GALINDO, ET AL.[17]. $c = r^{2e} + mn \bmod n^2$, where $(e, \lambda(n)) = 1$. Its one-wayness is reducible to factorization ($n = pq$, $p = q = 3 \bmod 4$).

KUROSAWA ET AL. [23]. $c = (r + \alpha/r)^e + mn (\text{mod} n^2)$, where e is a prime between $n/2$ to n and $(\alpha/p) = (\alpha/q) = -1$. Its one-wayness is reducible to factorization. In all these encryption scheme, the randomness r is recovered during decryption.

□ **End of chapter.**

Chapter 4

Identity-based Cryptography

Summary

The idea of Identity-based (ID-based) cryptography was proposed by Shamir[39] in 1984. In this new paradigm, users' identifying information such as email or IP address can be used as public key for encryption, signature or identification. ID-based cryptography avoid the need to link users to their public keys. Thus, it reduces system complexity and the cost for establishing and managing the public key authentication framework known as Public Key Infrastructure (PKI). In this chapter, we describe ID-based cryptography and review related results.

4.1 Introduction

In 1984, Shamir suggested a new idea for public key encryption scheme in which the public key can be an arbitrary string. The original motivation for such a scheme was to simplify certificate management. Since then several identity-based signature (IBS) and identity-based identification (IBI) schemes have been proposed. These include the Fiat-Shamir scheme [15], the schemes included in Shamir's paper introducing identity-bsaed cryptosystem[39], the Guillou-Quisquater scheme[20] and T. Okamoto scheme [30]. [2] provide detailed analysis on 14 existing IBI and IBS by providing a framework that reduces proving security of IBI and IBS schemes to proving security of an underlying SI scheme.

On the other hand, efficient Identity-based encryption (IBE) scheme did not appear until 2001, when Boneh and Franklin[6] proposed an IBE based on the bilinear Diffie-Hellman problem with respect to a pairing, such as the Weil pairing, and Cocks[11] based on the quadratic residuosity problem. Boneh and Franklin's scheme is considered much more efficient, and since then ID-based cryptography has been a very popular research topic.

Boneh and Franklin's scheme is secured in the random oracle model. Later, Canetti et. al. [36] describe a weaker model of security for IBE that they called the Selective-ID model. They proposed an IBE that is secure in this model without using the random oracle methodology. Boneh and Boyen [4] improve upon this result by describing an efficient scheme that is secure in the Selective-ID model. Recently, Boneh and Boyen [5] proposed another scheme that is fully secure without random oracles. Finally, a more efficient scheme is proposed by Waters[43].

4.2 Identity-based Encryption

An IBE is a four-tuple $(\mathsf{setup}, \mathsf{extract}, \mathsf{encrypt}, \mathsf{decrypt})$. setup takes security parameter λ to output system parameters param and master key pair $\mathsf{masterkey}$. $\mathsf{extract}$ takes param, $\mathsf{masterkey}$, and $ID \in \{0,1\}^*$, to output a user private key d. $\mathsf{encrypt}$ takes param, ID, and message M to output ciphertext C. $\mathsf{decrypt}$ takes param, C, private key d, to output message M. [6] defined *semantic security* of IBE as a form of IND-CPA security of the encryption system.

4.2.1 Notions of Security

Chosen Ciphertext Security. An identity-based encryption scheme $\mathcal{E}$ is semantically secure against an adaptive chosen ciphertext attack (IND-ID-CCA) if no polynomially bounded adversary $\mathcal{A}$ has a non-negligible advantage against the Challenger in the following IND-ID-CCA game:

Setup: The challenger takes a security parameter λ and runs the Setup algorithm. It gives the adversary the resulting system parameters params. It keeps the master-key to itself.

Phase 1: The adversary issues queries $q_1 \dots q_m$ where query q_i is one of:

- Extraction query $\langle \mathsf{ID}_i \rangle$. The challenger responds by running algorithm Extract to generate the private key d_i corresponding to the public key $\langle \mathsf{ID}_i \rangle$. It sends d_i to the adversary.
- Decryption query $\langle \mathsf{ID}_i, C_i \rangle$. The challenger responds by running algorithm Extract to generate the private key d_i corresponding to ID_i. It then runs algorithm Decrypt to decrypt the ciphertext C_i using the private key d_i. It sends the resulting plaintext to the adversary.

These queries may be asked adaptively, that is, each query q_i may depend on the replies to $q_1 \dots q_i$.

Challenge(Gauntlet): Once the adversary decides that Phase 1 is over it outputs two equal length plaintexts $M_0, M_1 \in \mathcal{M}$

and an identity ID on which it wishes to be challenged. The only constraint is that ID did not appear in any private key extraction query in Phase 1. The challenger picks a random bit $b \in \{0,1\}$ and sets $C = \mathsf{Encrypt}(\mathsf{params}, \mathsf{ID}, M_b)$. It sends C as the challenge to the adversary.

Phase 2: The adversary issues more queries $q_{m+1}, \ldots, q_n$ where query q_i is one of:

- Extraction query $\langle \mathsf{ID} \rangle$ where $\mathsf{ID}_i \neq \mathsf{ID}$. Challenger responds as in Phase 1.
- Decryption query $\langle \mathsf{ID}_i, C_i \rangle \neq \langle \mathsf{ID}, C \rangle$. Challenger responds as in Phase 1.

These queries may be asked adaptively as in Phase 1.

Guess: Finally, the adversary outputs a guess $b' \in \{0,1\}$. The adversary wins the game if $b = b'$.

We refer to such an adversary $\mathcal{A}$ as an IND-ID-CCA adversary. We define adversary $\mathcal{A}$'s advantage in attacking the scheme $\mathcal{E}$ as the following function of the security parameter λ (λ is given as input to the challenger): $Adv_{\mathcal{E},\mathcal{A}}(k) = |Pr[b = b'] - \frac{1}{2}|$. Using the IND-ID-CCA game we can define chosen ciphertext security for IBE schemes.

Definition 4.1. *An IBE system $\mathcal{E}$ is semantically secure against an adaptive chosen ciphertext attack if for any polynomial time IND-ID-CCA adversary $\mathcal{A}$ the function $Adv_{\mathcal{E},\mathcal{A}}(k)$ is negligible. As shorthand, we say that $\mathcal{E}$ is IND-ID-CCA secure.*

Note that the security requirements of an IBE was first formlized by Bohen and Franklin [6]. Interested readers may refered to the paper for detailed description.

4.2.2 Related Results

We review the IBE from Cocks [11].

- **Setup.** Generate two primes p and q, such that $p = q = 3 \bmod 4$, compute $N = pq$. $(mpk, msk) = ((n), (p, q))$. Define a hash function $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_n$
- **Extract.** Compute $Q = H_1(\cdots(H_1(\text{ID})\cdots)$, where hashing H_1 is applied repeatedly until the first result whose Jacobi symbol equals 1. Either Q or $-Q$ is in QR_n. Compute r such that $r^2 = Q$ or $r^2 = -Q$. The user secret key is r.
- **Encrypt.** Message $m \in \{-1, +1\}$: Choose $t, t' \in Z_n$ with $(\frac{t}{n}) = (\frac{t'}{n}) = m$. Send $c = (t + Q/t)(\text{mod}n)$ and $c' = (t' - Q/t')(\text{mod}n)$. as the ciphertext.

- **Decrypt.** Message $m=(\frac{c+2r \bmod n}{n})$ if $Q \in QR_n$ and $m = \frac{c'+2r \bmod n}{n})$

The scheme is IND-CPA-secure if quadratic residuosity problem is hard.

4.3 Identity-based Identification

An IBI scheme is a tuple $\mathcal{IBI}$=(Mkg, UKg, $\bar{\mathsf{P}}$, $\bar{\mathsf{V}}$). Mkg takes in security parameter λ and return master public and secret key pair (mpk,msk). Ukg on input msk, and an identity I, output user secret key usk. In the interactive identification protocol, $\bar{\mathsf{P}}$(initialized with usk, I) interact with $\bar{\mathsf{V}}$(initialized with I, mpk). The protocol ends when $\bar{\mathsf{V}}$ either accept or reject. [2] defined an IBI is *imp-atk-secure*, where $atk \in \{pa, aa, ca\}$ if it is secure against impersonation under passive, active or concurrent attack.

In this dissertation, we only consider three-move identification protocol of the following form.

1. $\bar{\mathsf{P}}$ sends a commitment t to $\bar{\mathsf{V}}$.
2. $\bar{\mathsf{V}}$ returns a challenge c which is randomly chosen from some set.
3. $\bar{\mathsf{P}}$ provides a response z.

4. Based on the input (mpk, I, t, c, z), $\bar{\mathsf{V}}$ output Accept or Reject.

4.3.1 Security notions

We consider three types of impersonation attack, namely, passive, active and concurrent attack, in the following game.

To model the attack scenario, we provide the adversary with the following oracles.

- $\mathcal{KEO}$. On input ID, output usk for the corresponding ID.
- $\mathcal{CO}$. On input ID, output a conversation transcript of the interactive protocol between $\bar{\mathsf{P}}$, $\bar{\mathsf{V}}$) for that identity.
- $\mathcal{PO}$ (Prover Oracle). On input ID, act as the prover $\bar{\mathsf{P}}$ to carry out the interactive identification protocol.

[Game IB-IMP]

1. Setup Phase: Dealer $\mathcal{D}$ runs $\mathsf{Mkg}(1^\lambda)$ to obtain (mpk, msk).
2. Probe Phase: Adversary $\mathcal{A}$ issue queries to the oracles. The queries can be interleaved.
3. At some point, $\mathcal{A}$ chooses a gauntlet ID, ID_G on which it wishes to impersonate and $\mathcal{A}$ act as the cheating prover now, trying to convince the verifier

The following restrictions applied. Passive attacker cannot query $\mathcal{PO}$. Active attacker can query $\mathcal{PO}$ only in a sequential manner. $\mathcal{A}$ wins the game if it can successfully convince the verifier and ID_G has never been input of $\mathcal{KEO}$.

Definition 4.2. *An ID-based identification scheme is ib-atk-imp-secure (atk $\in \{pa, aa, ca\}$ which stands for passive, active and concurrent) if no polynomial time adversary can win the above game with non-negligible probability.*

For detailed description of the security model for IBI, readers are recommended to [2].

4.4 Identity-based Signature

An ID-based signature (IBS) scheme is a four-tuple (Mkg, Ukg, IBSS, IBSV) specified as follow. Mkg, Ukg are the same as IBI. $(\sigma) \leftarrow \mathsf{IBSS}(\mathrm{ID}, mpk, usk, m)$ is a PPT algorithm which, on input ID, mpk, usk and message m, generate a signature σ. Accept/Reject $\leftarrow \mathsf{IBSV}(\mathrm{ID}, mpk, m, \sigma)$ is a PPT algorithm which, on input ID, signature σ, message m, output Accept or Reject.

An IBS should satisfy two properties, namely, completeness and soundness.

(**Completeness.**) A legitimate signature should be accepted. Formally, for all security parameter λ and $\forall \mathrm{ID} \in \{0,1\}^*$, $(mpk, msk) \in [\mathsf{Mkg}(1^\lambda)]$, and $usk \in [\mathsf{Ukg}(\mathrm{ID}, mpk, msk)]$, $\mathsf{Accept} \leftarrow \mathsf{IBSV}(\mathrm{ID}, mpk, m, \sigma)$ with overwhelming probability if $\sigma \leftarrow \mathsf{IBSS}(\mathrm{ID}, mpk, usk, m)$.

(**Soundness.**) An invalid signature should be rejected. Formally, for all security parameter λ and $\forall \mathrm{ID} \in \{0,1\}^*$, $(mpk, msk) \in [\mathsf{Mkg}(1^\lambda)]$, and $usk \in [\mathsf{Ukg}(\mathrm{ID}, mpk, msk)]$, $\mathsf{Reject} \leftarrow \mathsf{IBSV}(\mathrm{ID}, mpk, m, \sigma)$ with overwhelming probability if $\sigma \not\leftarrow \mathsf{IBSS}(\mathrm{ID}, mpk, usk, m)$.

4.4.1 Security notions

The accepted security notion for IBS is existential unforgeability against adaptive chosen ID and message attack (ib-uf-cma). We consider the following game.

To model the attack scenario, we provide the adversary with the following oracles.

- $\mathcal{KEO}$ defined before.
- *Signing Oracle($\mathcal{SO}$)*: $\sigma \leftarrow \mathcal{SO}(\mathrm{ID}, mpk, m)$. Upon inputs $\mathrm{ID} \in \{\mathrm{ID}\}$, mpk and message m, output a signature σ such that $\mathsf{Accept} \leftarrow \mathsf{IBSV}(\mathrm{ID}, mpk, m, \sigma)$.

  [Game IB-UF-CMA]

1. Setup Phase: Dealer $\mathcal{D}$ runs $\mathsf{Mkg}(1^\lambda)$ to obtain (mpk, msk).

2. Probe Phase: Adversary $\mathcal{A}$ issue queries to the oracles. At some point, $\mathcal{A}$ chooses a gauntlet ID, ID_G, to forge a signature with on any message of its choice. $\mathcal{A}$ cannot submit ID_G to $\mathcal{KEO}$ and it must be returned from $\mathcal{IO}$.

3. Delivery Phase: At the end, $\mathcal{A}$ submit a signature σ for message m of ID_G. m and ID_G pair must not be submitted to $\mathcal{SO}$ before. $\mathcal{D}$ outputs either Accept (if Accept $\leftarrow$ $\mathsf{IBSV}(\mathrm{ID}, mpk, m, \sigma)$) or Reject (otherwise).

The advantage of adversary is defined as the probability that Dealer output Accept.

Definition 4.3. *An IBS scheme (Mkg, Ukg, IBSS, IBSV) is* uf-cma-secure *if no PPT adversary has non-negligible advantage in Game IB-UF-CMA.*

□ **End of chapter.**

Chapter 5

Identity-Based Cryptography from Paillier System

Summary

In this chapter, we present several identity-based identification (IBI) schemes in the Paillier setting, and reduce their security to RSA-related assumptions in the random oracle model. The Fiat-Shamir paradigm can be used to turn them to identity-based signature (IBS) schemes. Next, we reformat Cocks'[11] IBE in the Paillier setting.

5.1 Identity-based Identification schemes in Paillier setting

In schemes below, hash function H mapping arbitrary string to random element of QR_{n^2} is used. However, in practice, how it can be implemented is unclear since deciding whether an element is a quadratic residue is hard without factorization. We adapt the technique by Cocks [11]. $H_c(\cdots(H_c(seed)\cdots) = w \bmod n^2$ until the hash output has Jacobi Symbol equal to 1. Note Jacobi Symbol can be computed without knowing the factoring of n. By our setting, either w or $-w$ is in QR_{n^2}.

5.1.1 Paillier-IBI

We present Pailler1,2-IBI, motivated by [32].

MKg: Generate two safe primes p and q, compute $n = pq$. Generate g of order αn where α is any integer. (mpk,msk)=((n,g),(p,q)).

UKg: For identity I, denote $Q = H(I)$, compute $(x, y) \in (\mathbb{Z}_n \times QR_n)$ such that $g^x y^n = Q(\bmod n^2)$.

$(\bar{\mathsf{P}},\bar{\mathsf{V}})$: (Commit, challenge, response)=(t, c, z) where $t =$

$\theta(g^r u^n \bmod n^2)$, for randomly generated r and u. c is random challenge. $z = (z_1, z_2) = (r - cx, uy^{-c}) \in (\mathbb{Z} \times \mathbb{Z}_n^*)$. Verify $t = \theta(Q^c g^{z_1} {z_2}^n (\mathrm{mod} n^2))$.

In paillier1-IBI, θ is the identity mapping while in Paillier2-IBI, θ is the random oracle.

Theorem 5.1. *Paillier1-IBI is imp-pa-secure if the RSA[n,n] assumption holds, in the Random Oracle Model.*

Theorem 5.2. *Paillier2-IBI is imp-aa,ca-secure if the RSA[n,n] assumption holds, in the Random Oracle Model.*

We outline three other IBIs in Paillier setting below.

5.1.2 CGGN-IBI

We present CGGN1,2-IBI , motivated by the scheme from Catalano et al.[10].

Key pairs: (*mpk*,*msk*):((*n*,*e*),(*p*,*q*)), where e is any public exponent relatively prime with $\phi(n)$. (usk_I)=(x,y)$\in$ ($\mathbb{Z}_n \times QR_n$) s.t. $H(I) = Q = (1+n)^x y^e (\bmod n^2)$. Also, denote by $g = 1+n$.

($\bar{\mathsf{P}}$,$\bar{\mathsf{V}}$): (Commit, challenge, response)=(t, c, z) where $t = \theta(g^r u^e \bmod n^2)$, for randomly generated r and u. c is random challenge $< e$. $z = (z_1, z_2) = (r - cx, uy^{-c}) \in (\mathbb{Z}_n \times \mathbb{Z}^*_{n^2})$. Verify $t = \theta(Q^c g^{z_1} {z_2}^e (\mathrm{mod} n^2))$.

In CGGN1-IBI, θ is the identity mapping while in CGGN2-IBI, θ is the random oracle.

Theorem 5.3. *CGHN1-IBI is imp-pa-secure if the RSA[n,e] assumption holds, in the Random Oracle Model.*

Theorem 5.4. *CGHN2-IBI is imp-aa,ca-secure if the RSA[n,e] assumption holds, in the Random Oracle Model.*

5.1.3 GMMV-IBI

GMMV-IBI is motivated by the scheme from Galindo, et al.[17].

Key pairs: (mpk,msk):$((n,e,K),(p,q))$. (usk_I)=$(x_k,y_k)\in (QR_n\times \mathbb{Z}_n)$ s.t. $x_k^{2e} + y_k n = H_k(I)$ for $k = 1, \ldots, K$. Denote $Q_k = H_k(I)$ for $k = 1, \ldots, K$.

$(\bar{\mathsf{P}},\bar{\mathsf{V}})$: (Commit, challenge, response)=(t, c, z) where $t = (r^{2e} + un(\mathrm{mod} n^2))$, for randomly generated r and u. $c = (c_1, \ldots, c_K)$ is random binary vector challenge. $z = (z_1, z_2) = (r \prod x_k^{-c_k}, ur^{-2e} - \sum c_k y_k x_k^{-2e}) \in (\mathbb{Z}_{n^2} \times \mathbb{Z}_n)$. Verify $t = (1 + n)^{z_2} {z_1}^{2e} \prod Q_k^{c_k} (\mathrm{mod} n^2)$

Theorem 5.5. *GMMV-IBI is imp-aa,ca-secure if Factorization is hard, in the Random Oracle Model.*

5.1.4 KT-IBI

KT-IBI is motivated by the scheme from Kurosawa et al.[23].

Key pairs: (mpk,msk):(N,α,K),(p,q), where $(\alpha/p) = (\alpha/q) = -1$.
(usk_I)=$(x_k,y_k)\in (QR_n \times \mathbb{Z}_n)$ s.t. $x_k + \alpha/x_k + y_k n = H_k(I)$ for $k = 1,\ldots,K$. Denote $Q_k = H_k(I)$ for $k = 1,\ldots,K$. Denote $A_k = x_k + \alpha/x_k$ and $B_k = x_k - \alpha/x_k$.

$(\bar{\mathsf{P}},\bar{\mathsf{V}})$: (Commit, challenge, response)=(t,c,z) where $t = r^2 + un(\mathrm{mod} n^2)$, for randomly generated r and u. $c = (c_1,\ldots,c_K)$ is random binary vector. $z = (z_1,z_2) = (r\prod B_k^{-c_k}, ur^{-2} - \sum c_k 2y_k A_k B_k^{-2}) \in (\mathbb{Z}_{n^2}\times\mathbb{Z}_n)$. Verify $t = (1+n)^{z_2} {z_1}^2 \prod (Q_k^2 - 4\alpha)^{c_k} (\bmod n^2)$.

Theorem 5.6. *KT-IBI is imp-aa,ca-secure if Factorization is hard, in the Random Oracle Model.*

Remarks: In using the Cocks technique, either H(I) or -H(I) is in QR_{n^2}. Prover should inform verifier which one is the case.

In the above protocols, we assume H(I) is the case.

5.1.5 Choice of g for Paillier-IBI

For Paillier1,2-IBI, there are several choice of g for the relation ((x,y),H(ID)) s.t. $H(ID) = g^x y^n (\mathrm{mod} n^2)$. The only restriction is order of g has to be multiple of n. For the simplest case, $g = 1 + n$ whose order is n can be used. The response z_1 in the identification protocol can then be computed in $\mathbb{Z}_n$. Moreover, $(1+n)^z = 1 + zn(\mathrm{mod} n^2)$ and this improves efficiency. We can also have the choice such that g is a generator of QR_{n^2}, in which order of g is $n\phi(n)$, unknown to public. This choice would affect the range of the randomly number during the identification protocol and is briefly explained as follow.

Commit. Randomly generate $r \in \mathbb{Z}_{\lfloor n^2/4 \rfloor}$, $n \in \mathbb{Z}_n^*$, compute $t = g^r u^n (\mathrm{mod} n^2)$.

Challenge. Randomly choose a challenge from $\mathbb{Z}_{q_c}$, where q_c is a prime smaller than the smallest prime factor of n.

Response. Compute $z_1 = r - cx \in \mathbb{Z}$ and $z_2 = uy^{-c}(\mathrm{mod} n)$.

Verify. Verify $t = \theta(H(ID)^c g^{z_1} {z_2}^n (\mathrm{mod} n^2))$.

In order to simulate the transcript, simulator first generate z_1 from $\{0, \cdots, \lfloor n^2/4 \rfloor\}$ and z_2 from $\mathbb{Z}_n$. Then it randomly generate c from $\mathbb{Z}_{q_c}$ and compute $t = H(ID)^c g^{z_1} {z_2}^B \bmod n^2$. To prove the simulated transcript is indistinguishable from the actual transcript, one has to consider the probability distribution of the responses. For z_2, it is obvious that the two distribution are both uniform. Consider the probability distribution of $P_{Z_1}(z_1)$ of the responses of the prover and the probability distribution $P_{Z'_1}(z'_1)$ according the the way simulator chooses z'_1. $P_{Z'_1}(z'_1)$ is uniformly distributed across $\{0, \cdots, \lfloor n^2/4 \rfloor\}$. It can be shown that the two distribution are indistinguishable if q_c is small enough.

We have in mind if p, q are 512-bit, then q_c is 80 bit.

5.2 Identity-based signatures from Paillier system

We can apply Fiat-Shamir transform [15] to the above IBI's and yield several IBS's. The resulting IBS's can be easily proven to be existentially unforgeable under adaptive chosen-message attack (*uf-cma-secure*) under the corresponding assumptions of the IBI's.

5.3 Cocks ID-based Encryption in Paillier Setting

We reformat Cocks' IBE [11] in Paillier setting so that the same setting of keys can be used for both IBS and IBE. Its security is equivalent to the security of Cock's original IBE.

Paillier-IBE

Setup: Generate two safe primes p and q, compute $n = pq$, and an element g whose order is multiple of n.

Extract: compute $Q = H_1(\cdots(H_1(\mathrm{ID})\cdots)$, where hashing H_1 is applied repeatedly until the first result whose Jacobi symbol equals 1. The secret key is $(flag, x, y)$ where (Case 1) $flag = 1$, $g^x y^{2n} = Q$, if $Q \in QR_n$; or (Case 2) $flag = -1$, $g^x y^{2n} = -Q$, if $-Q \in QNR_n$.

Encrypt: Message $m \in \{-1, +1\}$: Choose $t, t' \in Z_n$ with $(\frac{t}{n}) = (\frac{t'}{n}) = m$. Randomly generate r, r'. Send $c = g^r(t + Q/t)(\mathrm{mod} n^2)$ and $c' = g^{r'}(t' - Q/t')(\mathrm{mod} n^2)$.

Decrypt: If $flag = 1$, then compute message=$(\frac{c+2y^n \bmod n}{n})$. Else, compute message=$(\frac{c'+2y^n \bmod n}{n})$.

The following theorem can be proved easily.

Theorem 5.7. *Paillier-IBE is IB-OW-CPA secure if QRP Prob-*

lem is hard, in Random Oracle Model

There are well-known methods to convert an OW-CPA encryption to an IND-CCA encryption[3, 13, 32]. They can be used to convert Paillier-IBE to an IB-IND-CCA-secure IBE with multi-bit messages. We demonstrate by using OAEP[3]. Let m be a multi-bit message, G and H be secure hashing functions. Randomly generate r. Let $s = (m||0^\ell) \oplus G(r)$, $t = H(s) \oplus r$, ctxt be the bit-by-bit Paillier-IBE encryption of $(s||t)$. Then the scheme is IB-IND-CCA secure in ROM, provided the padding length ℓ is sufficiently large.

The particular conversion in Cocks [11] can also be used. But it comes without a formal proof of security.

We make the observation that Paillier-IBE (resp. Cocks' IBE) can be used as an *oblivious transfer (OT)*[16]. In a *1-2 OT*, Alice sends Bob two messages, Bob receives at most one, and Alice does not know which one. In a *chosen 1-2 OT*[28], Bob gets to choose which one he receives. Paillier-IBE (resp. Cock's IBE) can be used as a chosen 1-2 OT as follows: Alice and Bob both know n, and Bob may know its factoring. Bob generates π, $(\frac{\pi}{n}) = 1$, and sends it to Alice. Alice verifies $(\frac{\pi}{n}) = 1$, then encrypts multi-bit message m_0 to the case $\pi \in QR$ bit-by-bit, and she encrypts multi-bit message m_1 to the case $-\pi \in QR$

bit-by-bit, using Paillier-IBE (resp. Cocks' IBE). This is indeed a chosen 1-2 OT: Alice is assured Bob can only decrypt one message, but she does not know which one. But its bandwidth efficiency is poor.

□ **End of chapter.**

Chapter 6

Concluding Remarks

We have presented 4 different IBI schemes from Paillier system and extended them to IBS. We reduce their securities to RSA or Factoring Problem, in the random oracle model. Finally, we present Cocks IBE in Paillier setting with some discussions.

We recommend the following future research directions for this thesis.

Secure IBI without random oracle model. So far all of the results presented in this thesis are proven secure only under the random oracle model. As with ID-based encryption scheme, research direction could be to construct scheme secured in the standard model.

Extension to blind signature. Extension of the result to blind signature should be quite straight forward, especially for Paillier1-IBI.

Extension to ring signature and linkable ring signature.

Following the generic construction in [38], it is straight forward to construct identity-based ring signature from our results. Trying to construct identity-based linkable ring signature may be possible by following the technique from [42].

□ **End of chapter.**

Appendix A

Proof of Theorems

Summary

Proofs of the theorems are given in this section.

A.1 Proof of Theorems 5.1, 5.2

Proof of Theorem 5.1.

Our argument goes as follow. Suppose Paillier1-IBI is not imp-pa-secure. Then there exists an impersonator $\mathcal{I}$ which can impersonate the prover after observing a number of communication transcripts. We are going to show that if such $\mathcal{I}$ exists, then we can construct a simulator $\mathcal{S}$ which can solve the RSA[n,n] problem. This completed the proof of our theorem because we assume that no one can solve the RSA[n,n] problem. The existence of the impersonator $\mathcal{I}$ leads to the solution of the RSA[n,n] problem, which is a contradiction. The assumption that RSA[n,n] is hard is reasonable,since at present, no one can solve and it is widely believed to be hard.

Now we go through our argument by constructing such a simulator $\mathcal{S}$ which can solve the RSA[n,n] problem with the help of $\mathcal{I}$. We assume there is a fair dealer $\mathcal{D}$ which gives $\mathcal{S}$ a fair instance of the RSA[n,n] problem.

- Setup Phase. $\mathcal{S}$ received an instance of the RSA[n,n] problem from $\mathcal{D}$. That is, $\mathcal{S}$ is given (n, Q) and is asked to find y such that $y^n = Q \bmod n$. $\mathcal{S}$ then gives n and $g = 1 + n$ as *mpk* to impersonator $\mathcal{I}$.

- (Simulating the oracles.) Recalled that to model the attack scenario, $\mathcal{I}$ is given access to a number of oracles. Now $\mathcal{I}$, a passive attacker, can listen to communication transcript and ask for the secret key for any identity I. This is modeled by the oracle $\mathcal{CO}$ and $\mathcal{KEO}$ respectively. In the random oracle model, every hash function is also treated as oracle which the impersonator have access. The process that $\mathcal{S}$ handle the oracle query from $\mathcal{I}$ is called simulating the oracles or oracles simulation. Next we continue to show how $\mathcal{S}$ simulate the oracles for $\mathcal{I}$.

- H oracle. Suppose $\mathcal{I}$ makes q_H queries to the H oracle and let I_i denote the $i-th$ query. $\mathcal{S}$ randomly chooses r and return $Q = H(I_r)$. For other $i \neq r$, generate x_i, y_i and compute $H(I_i) = g^{x_i}{y_i}^n \bmod n^2$.

- $\mathcal{KEO}$. Suppose $\mathcal{I}$ query the secret key for I_i. $\mathcal{S}$ returns x_i, y_i. Suppose it query a new identity I', then set $H(I') = g^{x'}y'^n \bmod n^2$ and return (x', y'). This is called backpatch the random oracle H. The simulation failed if $\mathcal{I}$ query the secret key for I_r.

- $\mathcal{CO}$ is stimulated by randomly generate z_1, z_2, c and compute the commitment $t = H(I)^c g^{z_1} {z_2}^n \bmod n^2$. Return the transcript (t, c, z_1, z_2). It can be shown that statistical distance between the simulated transcript and actual transcript is negligible.

- (Gauntlet phase.) In the gauntlet phase, $\mathcal{I}$ chooses an identity I_g for impersonation. It is argued that $\mathcal{I}$ must choose one identity it has queried the H oracle. Otherwise the success probability is negligible. This argument is called the lunchtime argument. With probability $1/q_H$, $\mathcal{I}$ chooses $I_g = I_r$. If I_g is not I_r, then we also say the simulation fails.

- (Rewind Simulation.) Now suppose $\mathcal{I}$ can impersonate I_g

successfully. That is, $\mathcal{I}$ interactive with $\mathcal{S}$ in the identification protocol and is accepted. Let the communication transcript be $(t, c, (z_1, z_2))$. Now, since $\mathcal{I}$ is a computer program, we can reset the environment back to the point where $\mathcal{I}$ just issue the commitment t. At this point, $\mathcal{S}$ issue a challenge $c' \neq c$ and $\mathcal{I}$ impersonate successfully again. We let the transcript of the second-run be $(t, c', (z_1', z_2'))$. The process of resetting the environment (or state) of $\mathcal{I}$ is called rewind simulation.

- (Witness Extraction.) $\mathcal{S}$ can then compute some useful information from the two transcripts. This is called witness extraction. Assume $t = g^a b^n \bmod n^2$ and $Q = g^x y^n$.

$$
\begin{aligned}
g^a b^n &= g^{cx+z_1}(y^c z_2)^n \bmod n^2 \\
g^a b^n &= g^{c'x+z_1'}(y^{c'} z_2')^n \bmod n^2 \\
x &= (z_1' - z_1)/(c - c') \bmod n
\end{aligned}
$$

The last equation come from the fact that $[t]_g$ is unique modulo n and $[1]_g = 0$. $\mathcal{S}$ can compute y as follow.

$$
\begin{aligned}
t &= u^c (z_2)^n \bmod n \\
t &= u^{c'} (z_2')^n \bmod n \\
u^{c-c'} &= (z_2'/z_2)^n \bmod n
\end{aligned}
$$

Denote by s $z_2'/z_2 \bmod n$. $\mathcal{S}$ then compute (d, k_1, k_2) such that $d = \gcd(n, c - c')$ and $k_1 n + k_2(c - c') = d$. If $d \neq 1$, then $\mathcal{S}$ successfully factorize n (since $0 < c, c' < n$). Hence, $k_1 n + k_2(c - c') = 1$. $u = u^{k_1 n} u^{k_2(c-c')} = (u^{(k_1)}(s)^{k_2})^n \bmod n$. Thus, $y = u^{k_1} s^{k_2} \bmod N$.

- $\mathcal{S}$ compute y such that $y^n = H(I_g) \bmod n$ and successfully solved the RSA[n,n] problem.

- Probability of success depends on the simulation not failed. With probability $1/q_H$, $\mathcal{I}$ choose $I_g = I_r$ and it also implies that I_g is not input of $\mathcal{KEO}$. Since q_H is of polynomial complexity, probability of successful simulation is non-negligible.

Remarks: The proof required that gcd(n,c'-c)=1, thus, the challenge should be smaller than the smallest prime factor of n. By using $g = 1 + n$, efficiency can be improved. Also noted that order of $1 + n$ is n, the response z_1 will be in $\mathbb{Z}_n$ instead of in $\mathbb{Z}$.

Proof of Theorem 5.1. Now we can proceed to prove the imp-ca-security of Paillier2-IBI. It is in essence the same as Paillier1-IBI with the simulator now having to simulate the Prover Oracle. We only outline how the prover oracle is simulated here.

(Stimulating the prover oracle.) It is stimulated in Paillier2-

IBI by backpatching the θ oracle. Commitment t is randomly generated. After receiving the challenge c, backpatch $\theta(\ (H(I)^c g^{z_1} {z_2}^n))=t$. The response is ($z_1$,$z_2$).

A.2 Proof Sketch of Remaining Theorems

(*Proof Sketch of Theorems 5.3,5.4*)

- (Simulating the oracles.) $\mathcal{KEO}$, $\mathcal{CO}$ straight forward. $\mathcal{PO}$ is stimulated in a similar manner as in Paillier2-IBI.
- (Witness Extraction.) Given two conversation transcripts by $(t, c, (z_1, z_2))$ and $(t, c', (z_1', z_2'))$. Denote $H(I) = Q$. $y^e = Q \bmod N$ and $y^{c'-c} = (z_2/z_2') \bmod N$. Let $1 = k_1 e + k_2(c' - c)$, then $y = Q^{k_1}(z_2/z_2')^{k_2} \bmod n$. It successfully find the e-th root of Q modulo N and thus solves the RSA[n,e] problem.

(*Proof Sketch of Theorems 5.5*)

- (Simulating key extraction oracle.) Simulating $\mathcal{KEO}$ is straight forward by backpatching the H oracles.
- (Simulating Prover Oracle) GMMV-IBI employ witness indistinguishable technique, simulator possess one set of witness and the prover oracle can be simulated using the witness.

- (Witness Extraction.) Given two conversation transcripts denoted by $(t, c, (z_1, z_2))$ and $(t, c', (z_1', z_2'))$. $(z_1'/z_1)^2 = (\prod_{c_k=1} x_i / \prod_{c_k'=1} x_i)^2 \pmod{N}$. With probability 1/2, the two square roots differ and gcd of their difference leaks the factorization of N.

(*Proof Sketch of Theorem 5.6*)

- (Simulating the oracles.) $\mathcal{KEO}$ and $\mathcal{PO}$ are stimulated in a similar manner as in GMMV-IBI.

- (Witness Extraction.) Given two conversation transcripts $(t, c, (z_1, z_2))$ and $(t, c', (z_1', z_2'))$, it is straight forward to show $(z_1'/z_1)^2 = (\prod_{c_k=1} B_i / \prod_{c_k'=1} B_i)^2 \pmod{N}$. With probability 1/2, the two square roots differ and gcd of their difference leaks the factorization of N.

□ **End of chapter.**

Bibliography

[1] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology - EUROCRYPT 98*, pages 26–45. Springer-Verlag, 1998. Lecture Notes in Computer Science No. 1462.

[2] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. In *Advances in Cryptology - EUROCRYPT 04*, pages 268–286. Springer-Verlag, 2004. Lecture Notes in Computer Science No. 3027.

[3] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proc. 1st ACM Conference on Computer and Communications Security*, pages 62–73. ACM Press, 1993.

[4] D. Boneh and X. Boyen. Efficient selective-id secure iden-

tity based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 04*. Springer-Verlag, 2004. Lecture Notes in Computer Science.

[5] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 04*. Springer-Verlag, 2004. Lecture Notes in Computer Science.

[6] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology - CRYPTO 01*, pages 213–229. Springer-Verlag, 2001. Lecture Notes in Computer Science No. 2139.

[7] S. Brands. An efficient off-line electronic cash system based on the representation problem. *Technical Report CS-R9323m CWI*, 1993.

[8] E. Bresson, D. Catalano, and D. Pointcheval. A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In *Advances in Cryptology - ASIACRYPT 03*, pages 37–54. Springer-Verlag, 2003. Lecture Notes in Computer Science No. 2894.

[9] J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *Advances in*

Cryptology - CRYPTO 03, pages 126–144. Springer-Verlag, 2003. Lecture Notes in Computer Science No. 2729.

[10] D. Catalano, P. Gennaro, N.Howgrave-Graham, and P. Nguyen. Paillier's cryptosystem revisited. *ACM Communication and Computer Security -CCS'01, ACM pp.206-214, ACM*, 2001.

[11] C. Cocks. An identity based encryption scheme based on quadratic residues. In *Cryptography and coding*, pages 360–363. Springer-Verlag, 2001. Lecture Notes in Computer Science No. 2260.

[12] J. Cohen and M. Fischer. A robust and verifiable cryptographically secure election scheme. In *Proc. of the IEEE 26th Symposium on Foundations of Computer Science. Portland*, Dec. 1994.

[13] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Advances in Cryptology - CRYPTO 98*, pages 13–25. Springer-Verlag, 1998. Lecture Notes in Computer Science No. 1462.

[14] W. Diffie and M. E. Hellman. New directions in cryptography. In *IEEE Trans. Inform. Theory, IT-22(6)*, pages 644–654, Jan. 1976.

[15] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO 86*, pages 196–194. Springer-Verlag, 1986. Lecture Notes in Computer Science No. 263.

[16] M. J. Fischer, S. Micali, and C. Rackoff. A secure protocol for the oblivious transfer. In *Journal of Cryptology*, volume 9(3), pages 191–195, Mar. 1996.

[17] D. Galindo, S. Mollevi, P. Morillo, and J. Villar. A practical public key cryptosystem from paillier and rabin schemes. In *PKC'03*, pages 279–291. Springer-Verlag, 2003. Lecture Notes in Computer Science No. 2567.

[18] O. Goldreich. *Foundations of Cryptography.* Cambridge University Press, 2001.

[19] S. Goldwasser and S. Micali. Probabilistic encryption. In *Journal of Computer and System Sciences*, volume 28(2), pages 270–299, Apr. 1984.

[20] L. Guillou and J. J. Quisquater. A paradoxical identity-based signature scheme resulting from zero-knowledge. In *Advances in Cryptology - CRYPTO 88*, pages 216–231. Springer-Verlag, 1988. Lecture Notes in Computer Science No. 403.

[21] V. Klima. Finding md5 collisions v a toy for a notebook. *Cryptology ePrint Archive Report 2005/075*, 2005.

[22] D. E. Knuth. *The Art of Computer Programming, volume 2 - Seminumerical algorithms second edition.* Addison-Wesley, 1981.

[23] K. Kurosawa and T. Takagi. Some rsa-based encryption schemes with tight security reduction. In *Advances in Cryptology - ASIACRYPT 03*, pages 19–36. Springer-Verlag, 2003. Lecture Notes in Computer Science No. 2894.

[24] A. K. Lenstra and H. W. L. Jr. The development of the number field sieve. *volume 1554 of Lecture Notes in Mathematics. Springer Verlag*, 1993.

[25] W. Mao. *Modern Cryptography: Theory and Practice.* Prentice Hall PTR, 2003.

[26] U. M. Maurer and S. Wolf. The relationship between breaking the diffie-hellman protocol and computing discrete logarithms. In *Advances in Cryptology - CRYPTO 96*, pages 268–282. Springer-Verlag, 1999. Lecture Notes in Computer Science No. 1109.

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

[28] Y. Mu, J. Zhang, and V. Varadharajan. m out of n oblivious transfer. In *Proc. ACISP 2002*, pages 395–405. Springer-Verlag, 2002. Lecture Notes in Computer Science No. 2384.

[29] D. Naccache and J. Stern. A new public-key cryptosystem based on higher residues. In *Proc. 5th ACM Conference on Computer and Communications Security*, pages 59–66. ACM Press, 1998.

[30] T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *Advances in Cryptology - CRYPTO 92*, pages 231–253. Springer-Verlag, 1992. Lecture Notes in Computer Science No. 740.

[31] T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. In *Advances in Cryptology -*

EUROCRYPT 98, pages 308–318. Springer-Verlag, 1998. Lecture Notes in Computer Science No. 1403.

[32] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology - EUROCRYPT 99*, pages 223–238. Springer-Verlag, 1999. Lecture Notes in Computer Science No. 1592.

[33] S. C. Pohlig and M. E. Hellman. An improved algorithm for computing logarithms over gf(p) and its cryptographic significance. In *IEEE Trans. Inform. Theory, IT-24*, pages 106–110, Jan. 1978.

[34] J. M. Pollard. Monte carlo methods for index computation (mod p). In *Mathematics of Computation, 32(143)*, pages 918–924, July 1978.

[35] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. *MIT/LCS/TR-212, MIT Laboratory for Computer Science*, 1979.

[36] S. H. Ran Canetti and J. Katz. A forward-secure public-key encryption scheme. In *Advances in Cryptology - EUROCRYPT 03*. Springer-Verlag, 2003. Lecture Notes in Computer Science.

[37] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. In *Communications of the ACM, 21(2)*, pages 120–126, Feb. 1978.

[38] R. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *Advances in Cryptology - ASIACRYPT 01*, pages 552–565. Springer-Verlag, 2001. Lecture Notes in Computer Science No. 2248.

[39] A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology - CRYPTO 84*, pages 47–53. Springer-Verlag, 1984. Lecture Notes in Computer Science No. 196.

[40] V. Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology - EUROCRYPT 97*, pages 256–266. Springer-Verlag, 1997. Lecture Notes in Computer Science No. 1233.

[41] C. Studholme. *The Discrete Log Problem.* PhD Thesis, University of Toronto, 2001.

[42] P. P. Tsang and V. K. Wei. Short linkable ring signatures for e-voting, e-cash and attestation. In *Lecture Notes in Computer Science, Volume 3439*, pages 48–60, Mar. 2005.

[43] B. R. Waters. Efficient identity-based encryption without random oracles. *ePrint*, 2004(180), 2004.