Cryptanalysis of a Digital Signature Scheme of W. He

WONG, Chun Kuen

A Thesis Submitted in Partial Fulfilment

of the Requirements for the Degree of

Master of Philosophy

in

Information Engineering

June 2002

Abstract

In the new era of Digital Age, many daily life applications have been digitized — — signature scheme is not an exception. Signature forgery is of great concern for all signature schemes, regardless of whether they are paper signature schemes or digital signature schemes. Therefore, researchers are always finding new means to enhance the security of digital signature schemes.

Recently, to enhance security, W. He proposed a new digital signature scheme whose security is claimed to be based on the difficulty of solving two hard problems, factoring and discrete logarithm. There are two remarkable merits in the W. He signature scheme: First, users in the system can share a common modulus. Second, a user in the system needs to store only one number for her public key and only one number for her private key. As a result, the W. He signature scheme has an improvement in storage efficiency over the Brickell-McCurley signature scheme and the Okamoto signature scheme.

In this thesis, we present a cryptanalysis of the digital signature scheme of W. He, and prove that the digital signature scheme of W. He is not based on both the factoring and discrete logarithm problems.

摘要

踏入電子的世代，許多生活上的應用都被數碼化—簽章法也不例外。不論是對於人手簽章還是數位簽章，簽章的僞造都是必須關注的。因此，研究學者仍在不斷尋找加強數位簽章安全機制的新方法。

最近，W. He 提出了一個新的數位簽章算法來加強數位簽章的安全機制；並聲稱他的新數位簽章算法的安全機制是建基於同時求取兩個艱深的問題的解，分別是因數分解和離散對數。他的簽章法有兩個優越的地方。第一，系統的使用者可以共用同一個系統參數。第二， 系統的使用者只雖分別記存一個數字作爲私鑰、一個數字作爲公鑰。因此，若與 Brickell-McCurley 的簽章法和 Okamoto 的簽章法相比，W. He 的簽章法在記存的效率上有明顯的改善。

這篇論文將展示 W. He 簽章法的密碼分析、提出破解 W. He 簽章法的方法、 並證明 W. He 簽章法的安全機制並不是同時建基於因數分解和離散對數問題。

Acknowledgement

I would like to express my deepest gratitude to Prof. Victor K. W. Wei for his generous help with much of my research work. He has had the often difficult task of teaching me the discipline required to do research on cryptology. His marvelous intuition has helped me to channel my effort away from many dead ends that I would have spent months exploring. This thesis would not have been possible without his resourceful guidance and invaluable teaching.

I am happy to be affiliated to the Information Integrity Laboratory, where I developed friendship with my classmates. In addition, I would like to thank many other friends of mine for their support and encouragement.

Most of all, I owe a great debt to my parents who have given me persistent love and care.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Handwritten signature has long been widely used to provide authentication of messages, which is of fundamental importance in both the military and commercial arenas. Since it is believed that handwritten signature is unique for each person, signing on a document represents one's approval of its content. Moreover, that handwritten signature is easy to produce, easy to recognize, and difficult to forge makes it a practical and reliable form of authentication.

However, handwritten signature has some drawbacks, one of which is that a signer's handwritten signatures are the same for all documents. In other words, handwritten signature is independent of the document being signed. Thus, an adversary may be able to learn from studying examples of a signer's signatures and then successfully forge signatures for new desired documents. Although this attack may be a difficult task, it is not impossible. Another drawback of handwritten signature is that it is by its very nature not suitable for electronic commerce, in which everything is digital and can be processed quickly and efficiently to facilitate transactions.

In the new era of Digital Age, many daily life applications have been digitized —— signature scheme is not an exception. Digital signature, the digital counterpart of handwritten signature, is superior to handwritten signature in that it can eliminate the drawbacks of handwritten signature mentioned previously. The working principle and security of digital signature will be discussed in the next sections.

1.1 Origin of The First Digital Signature Scheme

The concept of "*digital signature*" was first proposed by Diffie and Hellman in their classic paper "*New Directions in Cryptography*" [1]. After that, the first practical realization of the concept was proposed by Rivest, Shamir and Adleman in [2]. This section gives an overview of the RSA signature scheme, which illustrates the general working principle of digital signature.

In the RSA signature scheme, each user (Alice) has a public key and a private key. The public key of Alice is known to all users in the system. On the other hand, the private key of Alice is kept secret and is known to Alice only. To sign a message, Alice uses her private key to produce a signature, which depends on both the message and Alice's private key. Then given the message and Alice's public key, any user in the system can use the public key of Alice to verify the correctness of the signature. The detail mathematical description of the RSA signature scheme is discussed in Chapter 3. Not only does the RSA be the first practical digital signature scheme, but it also has become the most popular digital signature scheme today.

While the above paragraph gives an overview of the RSA signature scheme, it

at the same time illustrates the general working principle of digital signature. As only the public key of a signer (say, Alice) is needed for signature verification, any user in the system can verify Alice's signatures. Moreover, since no one except Alice knows the value of her private key, signature proved to be valid for Alice's public key must have been signed by Alice. Furthermore, signatures of Alice are different for distinct messages since a signature depends on not only Alice's private key but also the message being signed. Therefore, even though a digital signature for a message can be duplicated easily, the replica cannot be used for other messages. Therefore, if in signature verification procedures a signature is proved to be Alice's signature for a message, then one can conclude that the signature for the message is indeed generated by Alice. As a result, authentication is achieved by the use of digital signature.

1.2 On the security of digital signature schemes

As the primary goal of signature is to provide authentication, signature forgery is of great concern.

While the security of handwritten signature lies in the difficulty of producing undetectable forged signatures, the security of digital signature depends on the intractability of mathematical hard problems. All digital signature schemes prevent signature forgery by making use of some well-known hard problems. For example, the security of the prevalent RSA digital signature scheme is based on the factoring problem. In the RSA signature scheme, if an attacker is able to factor a big composite integer, then he can achieve a total break of the system and forge signatures. (More specifically, if an attacker can solve the

factoring problem, then given the public key of a user the attacker is able to compute the corresponding private key of the user.) Known hard problems applicable to digital signature schemes include the factoring problem and the discrete logarithm problem.

From the previous discussion, we know that if one can solve the hard problem on which the security of a digital signature scheme is based, then he can break that digital signature scheme and forge signatures whenever he wants to. Therefore, in order to enhance the security of a digital signature scheme, the security of the digital signature scheme can be designed to be based on multiple (instead of single) hard problems. For example, Brickell, et al. [3] and Okamoto [4] have designed digital signature schemes whose security is based on both the factoring and discrete logarithm problems. Recently, W. He [5] proposed a new digital signature schemes whose security is claimed to be based on both the factoring and discrete logarithm problems. In this thesis, we present a cryptanalysis on the digital signature scheme of W. He, and prove that the digital signature scheme of W. He is not based on both the factoring and discrete logarithm problems.

1.3 Organization of the Thesis

The thesis is organized as follows. In Chapter 2, mathematical background that is involved in the thesis is introduced, including the Euler totient function, Fermat's little theorem, Euler's theorem, quadratic residue, Legendre symbol, Jacobi symbol, Blum integer, the factoring problem and the discrete logarithm problem.

Chapter 3 gives a survey on some remarkable digital signature schemes. This

includes the RSA signature scheme [2], ElGamal signature scheme [6], Schnorr signature scheme [7], Brickell-McCurley signature scheme [3], Okamoto signature scheme [4], Harn signature scheme [8], Shao signature scheme [9], and W. He signature scheme [5]. Throughout the chapter we will examine the merits, demerits and security of each digital signature scheme.

In Chapter 4 we present our cryptanalysis result on the digital signature scheme of W. He. It consists of two parts. In the first part, we introduce the W. He digital signature scheme, whose security is claimed to be based on both the factoring and discrete logarithm problems, and which has a remarkable merit : not only does it permit users in the system to share a common modulus, but it also allows a user in the system to store only one number for her public key and only one number for her private key. In the second part, we present a cryptanalysis of the W. He digital signature scheme and prove that its security is not based on both the factoring and discrete logarithm problem.

Finally, we conclude the thesis in Chapter 5.

Chapter 2

Mathematical Background

This chapter is a preliminary of the next chapters. In order to understand the cryptanalysis result presented in this thesis, a small amount of mathematical background and some notation are necessary, and they are presented in this chapter.

In the following discussion, each section will introduce a concept in number theory or cryptology by giving definitions, theorems and important properties. Since sections are interdependent, readers should start from the beginning to the last subsection. However, experts may choose to skip this chapter and proceed directly to the next chapter.

Proofs of theorems on number theory stated below can be found in [10].

2.1 Divisibility

Definition Let a, b be integers. If there exists an integer m such that $b = am$, then a *divides* b, which is denoted by $a|b$. The integer a is called the *divisor* of

b.

2.2 Prime

Definition Let p be an integer. If the only positive divisors of p are 1 and p, then p is a *prime* number (alternatively, p is prime). Otherwise, p is *composite.*

2.3 Modular arithmetic

Definition Let a, b be integers and $b \geq 1$. Then, there exists unique integers q, r such that $a = qb + r$ and $0 \leq r < b$. The remainder of the division of a by b, r, is denoted $a \bmod b$.

2.4 Congruence

Definition Let a, b, n be integers. If $n|(a-b)$, then a is *congruent* to b modulo n, denoted $a \equiv b \pmod{n}$. The integer n is called the *modulus* of the congruence.

2.5 Greatest Common Divisor

Definition Let a, b be integers. Then, the *greatest common divisor* of a and b, $gcd(a, b)$ is the largest positive integer that divides both a and b.

2.6 Integers modulo n

Definition Z_n denotes the set of integers $\{0, 1, 2, ..., n-1\}$. Addition, subtraction and multiplication in Z_n are done modulo n.

Definition $Z_n^* = \{a \in Z_n | gcd(a, n) = 1\}$ denotes the multiplicative group of Z_n.

2.7 Inverse

Definition Let $a \in Z_n$. If there exists an integer $x \in Z_n$ such that $ax \equiv 1 \pmod{n}$, then x is called the *multiplicative inverse* of a modulo n, denoted by a^{-1}. Otherwise, a does not have a multiplicative inverse.

Theorem 2.7.1 *If $gcd(a, n) = 1$, then a^{-1} modulo n exists.*

2.8 Division in Z_n

Definition Let a,$b \in Z_n$. Then, division of a by b modulo n is defined iff b^{-1} exists. Moreover, the division of a by b is defined as the product of a and b^{-1} modulo n.

2.9 Order of element

Definition Let $a \in Z_n^*$. Then, the *order* of a is the smallest positive integer k such that $a^k \equiv 1 \pmod{n}$.

2.10 Euclidean Algorithm

Theorem 2.10.1 *Let a, b be positive integers. Without loss of generality, assume $a \geq b$. Then, $gcd(a, b) = gcd(b, a \bmod b)$.*

Algorithm 2.10.1 (Euclidean Algorithm)

INPUT: positive integers a, b with $a \geq b$.

OUTPUT: $gcd(a, b)$.

1. Compute $r = a \bmod b$.
2. Set $a = b$ and then $b = r$.
3. Repeat steps 1 and 2 until $b = 0$.
4. Output the value of a.

2.11 Extended Euclidean Algorithm

The Euclidean Algorithm can be extended to find two integers (x, y) such that $ax + by = gcd(a, b)$, given integers a and b.

If $gcd(a, b) = 1$, then in Z_b, a^{-1} exists. By using the extended Euclidean algorithm, one can find (x, y) such that $ax + by = 1$. Thus, one can find x such that $ax \equiv 1 \pmod{b}$. Note that x is equal to a^{-1} in Z_b^* by the definition of inverse. Therefore, given n, $a \in Z_n^*$, one can use the extended Euclidean algorithm to find the multiplicative inverse a^{-1} in Z_n^*.

Algorithm 2.11.1 (Extended Euclidean Algorithm)

INPUT: positive integers a, b with $a \geq b$.

OUTPUT: $d = gcd(a, b)$ and integers x,y satisfying $ax + by = d$.

1. Set $x_1 = 0, x_2 = 1, y_1 = 1, y_2 = 0$.
2. Compute $q = \lfloor \frac{a}{b} \rfloor$, $r = a \bmod b$, $x = x_2 - qx_1$, $y = y_2 - qy_1$, where $\lfloor x \rfloor$ denotes the largest integer that is smaller than or equal to x.
3. Set $a = b, b = r, x_2 = x_1, x_1 = x, y_2 = y_1, y_1 = y$.
4. Repeat steps 2 and 3 until $b = 0$.
5. Set $d = a, x = x_2, y = y_2$ and output (d, x, y).

2.12 Chinese Remainder Theorem

Theorem 2.12.1 *Let $a_1, a_2, ..., a_k$ be integers. Let $n_1, n_2, ..., n_k$ be integers such that $gcd(n_i, n_j) = 1$ for all $i \neq j$. Then, the system of simultaneous congruences:*

$$\begin{cases} x \equiv a_1 \pmod{n_1} \\ x \equiv a_2 \pmod{n_2} \\ \vdots \\ x \equiv a_k \pmod{n_k} \end{cases}$$

has a unique solution $x = \sum_{i=1}^{k} a_i N_i M_i \bmod n$ in Z_n where $n = n_1 n_2 ... n_k$, $N_i = \frac{n}{n_i}$ and $M_i = N_i^{-1} \bmod n_i$.

2.13 Relatively Prime

Definition Let a, b be integers. Then, a, b are *relatively prime* if $gcd(a, b) = 1$.

2.14 Euler Totient Function

Definition The Euler totient function, $\phi(n)$, is defined as the number of integers in the interval $[1, n)$ which are relatively prime to n.

Properties:

1. If p is a prime number, then $\phi(p) = p - 1$.
2. If $gcd(m, n) = 1$, then $\phi(mn) = \phi(m)\phi(n)$.
3. If $n = p_1^{e_1} p_2^{e_2} ... p_k^{e_k}$ where $p_1, p_2, ..., p_k$ are distinct prime numbers, then $\phi(n) = n(1 - 1/p_1)(1 - 1/p_2)...(1 - 1/p_k)$.

2.15 Fermat's Little Theorem

Theorem 2.15.1 (Fermat's little Theorem) *Let a be an integer, p be a prime number. If $gcd(a, p) = 1$, then $a^{p-1} \equiv 1 \pmod{p}$.*

Remarks:

1. Given an integer a and a prime number p, a^{p-1} may not be congruent to 1 modulo p. Nevertheless, a^p is congruent to a modulo p for all integers a.
2. By Fermat's little theorem, if $r \equiv s \pmod{p-1}$, then $a^r \equiv a^s \pmod{p}$ for all integers a given that p is prime. Therefore, when working modulo a prime number p, exponents can be reduced modulo $(p - 1)$.

2.16 Euler's Theorem

Theorem 2.16.1 (Euler's Theorem) *Let n be an integer and $n \geq 2$. If $a \in Z_n^*$, then $a^{\phi(n)} \equiv 1 \pmod{n}$.*

Remarks:

1. A special case of the Euler's theorem is the Fermat's little theorem.

2.17 Square root

Definition Let $b \in Z_n$. If $x \in Z_n^*$ satisfies $x^2 \equiv b \pmod{n}$, then x is a *square root* of b modulo n.

2.18 Quadratic residue

Definition Let $b \in Z_n^*$. If there exists $x \in Z_n^*$ such that $x^2 \equiv b \pmod{n}$, then b is a *quadratic residue* modulo n. If no such x exists, then b is a *quadratic non-residue* modulo n.

Definition Q_n denotes the set of all quadratic residues modulo n. $\overline{Q}_n$ denotes the set of all quadratic non-residues modulo n.

Remarks:

1. 0 is not a member of Q_n nor $\overline{Q}_n$, since $0 \notin Z_n^*$.

Properties:

1. Let p be an odd prime. Then, there are exactly $(p-1)/2$ quadratic residues modulo p and $(p-1)/2$ quadratic non-residues modulo p. In other words, $|Q_p| = \frac{p-1}{2}$ and $|\overline{Q}_p| = \frac{p-1}{2}$.
2. Let $n = pq$ where p, q are distinct odd primes. Then, there are exactly $(p-1)(q-1)/4$ quadratic residues modulo p, and $3(p-1)(q-1)/4$ quadratic non-residues modulo p.
3. Let $b \in Q_p$ where p is an odd prime. Then, b has exactly two square roots modulo p, one of which is also in Q_p.

2.19 Legendre Symbol

The Legendre symbol is a useful tool for testing quadratic residuosity.

Definition Let p be an odd prime and a be an integer. The Legendre symbol, $(\frac{a}{p})$, is defined as:

$$(\frac{a}{p}) = \begin{cases} 0 & \text{if} \quad p|a \\ 1 & \text{if} \quad a \in Q_p \\ -1 & \text{if} \quad a \in \overline{Q}_p \end{cases}$$

Properties: Let p be an odd prime and a, b be integers. Then, $(\frac{a}{p})$ has the following properties:

1. $(\frac{a}{p}) = a^{(p-1)/2} \bmod p$. This formula is a useful tool for checking whether or not an integer a is a quadratic residue modulo a prime p.
2. $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$. Therefore, if $a \in Z_p^*$, then $(\frac{a^2b}{p}) = (\frac{b}{p})$ since $(\frac{a^2}{p}) = (\frac{a}{p})(\frac{a}{p}) = (\frac{a}{p})^2 = 1$.
3. If $a \equiv b \pmod p$, then $(\frac{a}{p}) = (\frac{b}{p})$.
4. $(\frac{1}{p}) = 1$
5. $(\frac{2}{p}) = (-1)^{(p^2-1)/8}$. Thus, $2 \in Q_p$ if $p \equiv 1$ or $7 \pmod 8$, and $2 \in \overline{Q}_p$ if $p \equiv 3$ or $5 \pmod 8$
6. If q is an odd prime distinct from p, then $(\frac{p}{q}) = (\frac{q}{p})(-1)^{(p-1)(q-1)/4}$. This is called the law of quadratic reciprocity. Hence, if $p \equiv 3 \pmod 4$ and $q \equiv 3 \pmod 4$, then $(\frac{p}{q}) = -(\frac{q}{p})$. Otherwise, $(\frac{p}{q}) = (\frac{q}{p})$.

2.20 Jacobi Symbol

Jacobi symbol is the generalization of Legendre symbol in that n is not necessarily prime.

Definition Let $n > 1$ be an odd integer. Suppose $n = p_1^{e_1} p_2^{e_2} ... p_k^{e_k}$ where $p_1, p_2, ..., p_k$ are distinct prime numbers and $e_1, e_2, ..., e_k$ are positive integers. Then, the Jacobi symbol, $(\frac{a}{n})$, is defined as:

$$(\frac{a}{n}) = (\frac{a}{p_1})^{e_1} (\frac{a}{p_2})^{e_2} ... (\frac{a}{p_k})^{e_k}$$

Properties: Let $n > 1, m > 1$ be odd integers and a, b be integers. Then, $(\frac{a}{n})$ has the following properties:

1. Since a Jacobi symbol is a product of Legendre symbols, $(\frac{a}{n}) = 0, 1,$ or -1. Moreover, $(\frac{a}{n}) = 0$ iff $gcd(a, n) \neq 1$.
2. Since $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$ (where p is prime) for Legendre symbols, we have a similar property for Jacobi symbol : $(\frac{ab}{n}) = (\frac{a}{n})(\frac{b}{n})$
3. $(\frac{a}{mn}) = (\frac{a}{m})(\frac{a}{n})$.
4. Since for Legendre symbol $(\frac{a}{p}) = (\frac{b}{p})$ if $a \equiv b \pmod{p}$ where p is an odd prime, we have a similar property for Jacobi symbol : if $a \equiv b \pmod{n}$, then $(\frac{a}{n}) = (\frac{b}{n})$.
5. Since $(\frac{1}{p}) = 1$, it is obvious that $(\frac{1}{n}) = 1$.
6. $(\frac{-1}{n}) = (-1)^{(n-1)/2}$. Thus, if $n \equiv 1 \pmod{4}$, $(\frac{-1}{n}) = 1$. Otherwise, $(\frac{-1}{n}) = -1$
7. $(\frac{2}{n}) = (-1)^{(n^2-1)/8}$. Thus, if $n \equiv 1$ or $7 \pmod{8}$, $(\frac{2}{n}) = 1$. Otherwise, $(\frac{2}{n}) = -1$

8. $(\frac{m}{n}) = (\frac{n}{m})(-1)^{(m-1)(n-1)/4}$. Hence, if $m \equiv 3 \pmod 4$ and $n \equiv 3 \pmod 4$, then $(\frac{m}{n}) = -(\frac{n}{m})$. Otherwise, $(\frac{m}{n}) = (\frac{n}{m})$.

Remarks:

1. It is easy to see that if the Jacobi symbol $(\frac{a}{n}) = -1$, then a is a quadratic non-residue modulo n. The interesting case is $(\frac{a}{n}) = 1$, whence no conclusion can be drawn regarding whether or not a is a quadratic residue modulo n. Obviously, if a is a quadratic residue modulo n, then $(\frac{a}{n}) = 1$. However, if $(\frac{a}{n}) = 1$, it does not imply that a is a quadratic residue modulo n. For example, observe that $(\frac{7}{143}) = 1$ but $7 \notin Q_{143}$

2.21 Blum Integer

Definition Let n be an integer. If $n = pq$ where p, q are distinct prime numbers, each of which is congruent to 3 modulo 4, then n is called a *Blum integer.*

Properties: Let $n = pq$ be a Blum integer. Let a be an integer. Then the following properties hold:

1. $(\frac{-1}{n}) = (\frac{-1}{p})(\frac{-1}{q}) = (-1)(-1) = 1$.
2. Since $(\frac{-a}{p}) = -(\frac{a}{p})$ and $(\frac{-a}{q}) = -(\frac{a}{q})$, we have $(\frac{-a}{n}) = (\frac{a}{n})$.
3. If $x,y \in Z_n^*$ such that $x^2 \equiv y^2 \pmod n$ and neither $x \equiv y \pmod n$ nor $x \equiv -y \pmod n$, then $(\frac{x}{n}) = -(\frac{y}{n})$.
4. If $a \in Q_n$, then a has four square roots modulo n, exactly one of which is also in Q_n.

2.22 The Factoring Problem

Definition Given a positive integer n, find its prime factorization; that is, find distinct prime numbers $p_1, p_2, ..., p_k$ and positive integers $e_1, e_2, ..., e_k$ such that $n = p_1^{e_1} p_2^{e_2} ... p_k^{e_k}$.

The factoring problem is one of the oldest problems in number theory. If an integer n has less than 10 bits (i.e. $n < 1024$), then it is easy to solve the factoring problem of n by simple trial division. However, if an integer n is large and has 2048 bits, then the factoring problem of n is so computationally intractable that the trial division approach would take millions of years to complete. Of course, integer factorization has advanced significantly in the last decades, and there are advanced methods for solving the factoring problem much superior than the trial division approach. Well-known factoring algorithms include Pollard's rho algorithm, Pollard's $p-1$ algorithm, quadratic sieve, and number field sieve. Among these advanced factoring algorithm, number field sieve is the fastest and has a heuristic asymptotic time estimate of $e^{(1.923+O(1))(\ln n)^{1/3}(\ln \ln n)^{2/3}}$ [11]. Nevertheless, the factoring problem of a 1024-bit-integer still remains a *"hard"* problem today, and the security of many cryptographic techniques depends upon the intractability of the factoring problem. In particular, the security of the RSA [2] signature scheme is based on the factoring problem, and in practice, 1024-bit integers are commonly used.

2.23 The Discrete Logarithm Problem

Definition Given a prime number p, integers $\alpha, \beta \in Z_p^*$, find an integer x, $1 \leq x \leq p-1$, such that $\alpha^x \equiv \beta \pmod{p}$.

Remarks: The discrete logarithm problem can be generalized to any finite cyclic group [10].

The discrete logarithm, like the factoring problem discussed in the last subsection, is a hard problem in number theory. So far, three remarkable algorithms for solving the discrete logarithm problem have been proposed: the linear sieve, the Guassian integer scheme, and the number field sieve. Despite that there have been significant advancement in algorithms for solving the discrete logarithm problem, discrete logarithm in Z_p where p is a large prime still remains a *"hard"* problem today.

The security of many cryptographic techniques depends upon the intractability of the discrete logarithm problem. The ElGamal signature scheme [6] is a well-known example, and in practice 1024-bit prime numbers are commonly used.

2.24 One-way Hash Function

Definition A one-way hash function h maps an input x of arbitrary finite length to an output $h(x)$ of fixed bitlength n, with the following properties:

1. For any given y, it is computationally infeasible to find x with $h(x) = y$.
2. For any given x, it is computationally infeasible to find $\overline{x} \neq x$ such that $h(\overline{x}) = h(x)$.

In general, $h(x)$ is much smaller than x; for example, x might be one megabyte, whereas $h(x)$ might be only 128 bits. Thus, in practice, hash functions are used in conjunction with digital signature schemes : a message m is hashed first, and then the hash value $h(m)$ is signed in place of the original message. This use of hash functions will reduce both the computation time of signing process and the size of the signature. However, the use of hash function in digital signature creates a problem: on one hand, to avoid signature forgery, signatures for different messages should be distinct, but on the other hand, for all one-way hash functions there must exist at least two distinct messages that have the same hash value, since a one-way hash function is mapping a message of arbitrary length into a fixed bitlength output. Thus, collision cannot be avoided entirely, since in general the number of possible messages exceeds the number of possible outputs of the hash function. Therefore, in property 2 of the definition of a one-way hash function, "computationally infeasible" is the best that a one-way hash function can do to avoid collision of hash values.

The design of a one-way function is by no means easy. Moreover, it turns out that many one-way hash functions are insecure and have been broken. For example, in [12], Schnorr proposed a one-way hash function based on the discrete Fourier transform. However, it was broken by [13, 14]. After that, Schnorr proposed a revised version called *FFT-HashII* in [15], which was then broken in [16]. Tillich proposed another one-way hash function called SL_2 in [17], however, it is insecure and broken by [18].

Chapter 3

Survey of digital signature schemes

Before presenting our cryptanalysis of the digital signature scheme in the next chapter, we in this chapter review some notable digital signature schemes related to our cryptanalysis result. This chapter gives a survey of some remarkable digital signature schemes, including the RSA signature scheme [2], ElGamal signature scheme [6], Schnorr signature scheme [7], Brickell-McCurley signature scheme [3], Okamoto signature scheme [4], Harn signature scheme [8], Shao signature scheme [9], and W. He signature scheme. The merits, demerits and security of these digital signature schemes are discussed.

3.1 The RSA signature scheme

Since the advent of public key cryptography by Diffie and Hellman [1], the first digital signature scheme invented was the RSA digital signature scheme [2]. This

section introduces the RSA digital signature scheme and discusses the security of the RSA digital signature scheme.

3.1.1 Key generation in the RSA signature scheme

A user in the system generates her private key d and public key (e, N) as follows:

1. Randomly choose two distinct prime numbers p and q.
2. Compute $N = pq$ and $\phi = (p-1)(q-1)$.
3. Randomly select an integer e such that $1 < e < \phi$ and $gcd(e, \phi) = 1$.
4. Use the extended Euclidean algorithm to find d such that $1 < d < \phi$ and $ed \equiv 1 \pmod{\phi}$.
5. Finally, the private key is d, and the public key is (e, N).

Remark: In the RSA digital signature scheme, N is called the public modulus, and e is called the public exponent.

3.1.2 Signature generation in the RSA signature scheme

Using her private key d, a user (Alice) in the system can generate her signature on message m as follows:

1. Compute $s = m^d \bmod N$.
2. Finally, the signature of Alice for message m is s.

3.1.3 Signature verification in the RSA signature scheme

Given a user (Alice)'s signature s for message m, another user (Bob) can use Alice's public key (e, N) to verify Alice's signature s for message m as follows:

1. Compute $\hat{m} = s^e \bmod N$.
2. Accept the signature iff $\hat{m} \equiv m \pmod{N}$.

3.1.4 On the security of the RSA signature scheme

This subsection considers several attacks on the RSA signature scheme, each of which has various requirements and effectiveness.

A. Total break (Integer Factorization Attack)

Suppose an adversary is able to factor the public modulus N of a user (Alice), i.e. the adversary can find distinct prime numbers p, q such that $N = pq$, given N is the RSA public modulus of user Alice. Then, the adversary can compute $\phi = (p-1)(q-1)$, and uses the extended Euclidean algorithm to determine the private key of user Alice by solving $ed \equiv 1 \pmod{\phi}$. Thus, *Integer Factorization Attack* leads to a total break of the system.

Remark: As discussed in section 2.22, factoring a large composite number is a hard problem. Therefore, this attack can be prevented by using a large public modulus N. In practice, a 1024-bit public modulus is hard to factor.

B. Existential Forgery

The existential forgery proceeds as follows. First, an adversary chooses a value for s. Next, given the public key (e, N) of user Alice, the adversary compute $m = s^e \bmod N$. Then, s is Alice's signature for the message m.

Remark: One way to counteract existential forgery in the RSA signature scheme

is to limit the set of messages that can be signed by the system users. For example, all message which are allowed to be signed must have their last several bits conform to a specific pattern.

C. Multiplicative Attack

Suppose that Alice is a user in the system and an adversary possesses Alice's signatures s_1 for message m_1 and s_2 for message m_2. Then, the adversary can compute $s_3 = s_1 s_2 \bmod N$ where N is the RSA public modulus of Alice. Note that s_3 is a valid signature of Alice for the message $m_3 = m_1 m_2 \bmod N$, since $s_3 \equiv s_1 s_2 \equiv m_1^e m_2^e \equiv (m_1 m_2)^e \pmod{N}$, where e is the RSA public exponent of Alice.

Remark: The above attack can be prevented by applying a one-way hash function (say, h) to the message before signature generation. Then, even though an adversary can compute s_3, he still has to invert the one-way hash function h in order to compute the new message $m_3 = h^{-1}(h(m_1)h(m_2))$.

3.2 The ElGamal signature scheme

In the previous section, we see that the security of the RSA signature scheme is based on the factoring problem. In this section, we introduce the ElGamal signature scheme [6], a classical digital signature scheme whose security is based on the discrete logarithm problem.

3.2.1 Key generation in the ElGamal signature scheme

A user in the system proceeds as follows to generate her private key a and public key (p, α, y):

1. Choose a random, large prime number p.
2. Randomly choose a primitive element α in Z_p^*, i.e. the order of α in the multiplicative group Z_p^* is $(p-1)$.
3. Select a random integer a, $1 \leq a \leq p-2$.
4. Compute $y = \alpha^a \bmod p$.
5. Finally, the private key is a, and the public key is (p, α, y).

Remark: In the ElGamal digital signature scheme, p is called the modulus, and α is called the base.

3.2.2 Signature generation in the ElGamal signature scheme

To generate her signature on a message m, user Alice in the system can use her private key a to perform the following operations:

1. Randomly choose an integer k, $1 \leq k \leq p-2$, such that $gcd(k, p-1) = 1$.
2. Compute $r = \alpha^k \bmod p$.
3. Use the extended Euclidean algorithm to compute k^{-1}.
4. Compute $s = k^{-1}(m - ar) \bmod (p-1)$.
5. Finally, the signature of Alice for message m is (r, s).

3.2.3 Signature verification in the ElGamal signature scheme

Given a user (Alice)'s signature (r, s) for message m, another user (Bob) can use Alice's public key (p, α, y) to verify Alice's signature (r, s) for message m as

follows:

1. Compute $v_1 = y^r r^s \bmod p$.
2. Compute $v_2 = \alpha^m \bmod p$.
3. Accept the signature iff $v_1 = v_2$.

3.2.4 On the security of the ElGamal signature scheme

Several attacks on the ElGamal signature scheme, each of which has various requirements and effectiveness, are discussed in this subsection.

A. Total break (Discrete Logarithm Attack)

Given user Alice's public key (p, α, y), if an adversary is able to compute discrete logarithm of the modulus y with respect to the base α in Z_p, then the adversary can compute a such that $y \equiv \alpha^a \pmod p$. Since a is the private key of user Alice, the adversary can then forge signatures on behalf of Alice. Thus, the *Discrete Logarithm Attack* constitutes a total break of the system.

Remark: As discussed in section 2.23, discrete logarithm is a hard problem. Therefore, this attack can be prevented by using a large modulus p. In practice, for a 1024-bit public modulus, it is hard to compute discrete logarithm.

B. Existential Forgery

An existential forgery attack proceeds as follows. Given the public key (p, α, y) of user Alice, an adversary randomly selects integers (u, v) such that $gcd(v, p-1) = 1$. Next, the adversary computes $r = \alpha^u y^v \bmod p$ and $s = -rv^{-1} \bmod (p-1)$. Then, (r, s) is Alice's signature for the message $m = su \bmod (p-1)$.

Remark: The above attack can be prevented by applying a one-way hash function (say, h) to the message before signature generation. Then, even though an adversary can compute (r, s) as mentioned above, he still has to invert the one-way hash function h in order to compute the new message $m = h^{-1}(su)$.

C. Attack on signatures having the same k

Suppose that an adversary possesses two signatures of a signer and the value of k of the two signatures are the same. Then, we have $s_1 = k^{-1}(m_1 - ar) \bmod (p-1)$ and $s_2 = k^{-1}(m_2 - ar) \bmod (p-1)$, where $r = \alpha^k \bmod p$, (p, α, y) is the public key of a user (say, Alice) and a is the private key of Alice. Therefore, the adversary can easily determine the private key of the signer by computing $a = (m - s_1 k)r^{-1} \bmod (p-1)$.

Remark: The use of a one-way hash function still cannot prevent this attack. Thus, it is important for the signer to use a different value of k for each signature generation.

D. Acceptable range of r

Consider the following attack: Suppose that an adversary has one valid signature s of a user (Alice) on a message m. Then, the adversary selects a new message $\hat{m}$ of its choices, computes $u = m\hat{m}^{-1} \bmod (p-1)$ and $\hat{s} = su \bmod (p-1)$. Next, by using Chinese Remainder Theorem discussed in section 2.12 , the adversary computes $\hat{r}$ such that $\hat{r} \equiv ru \pmod{p-1}$ and $\hat{r} \equiv r \pmod{p}$. Thus, the adversary has computed a signature $(\hat{r}, \hat{s})$ for the desired message $\hat{m}$.

Remark: The above attack cannot be prevented by using one-way hash functions. To obviate the above attack, a constraint on r, $1 \leq r \leq p-1$, must be incorporated into the signature verification procedure.

3.3 The Schnorr signature scheme

This section introduces the Schnorr signature scheme [7], which is a well-known variant of the ElGamal signature scheme.

3.3.1 Key generation in the Schnorr signature scheme

A user in the system proceeds as follows to generate her private key a and public key (p, q, v, α, h):

1. Choose two random, large prime numbers p and q such that $q|(p-1)$.
2. Randomly select an element α in Z_p, with order q.
3. Choose a one-way hash function $h : Z_q \times Z \mapsto (0, ..., 2^t - 1)$, where t is a security parameter.
4. Randomly choose an integer $a \in \{1, 2, ..., q\}$ and compute $v = \alpha^{-a} \bmod p$.
5. Finally, the private key is a, and the public key is (p, q, v, α, h).

3.3.2 Signature generation in the Schnorr signature scheme

Using her private key a, a user (Alice) in the system can generate her signature on message m as follows:

1. Randomly select $r \in \{1, ..., q-1\}$, and compute $x = \alpha^r \bmod p$.
2. Compute $e = h(x, m) \in \{0, ..., 2^t - 1\}$.

3. Compute $y = r + ae \pmod{q}$.
4. Finally, the signer's signature for message m is (e, y).

3.3.3 Signature verification in the Schnorr signature scheme

Given a user (Alice)'s signature (e, y) for message m, another user (Bob) can use Alice's public key (p, q, v, α, h) to verify Alice's signature (e, y) for message m as follows:

1. Compute $\hat{x} = \alpha^y v^e \bmod p$.
2. Compute $\hat{e} = h(\hat{x}, m)$.
3. Accept the signature iff $\hat{e} = e$.

3.3.4 Discussion

In the Schnorr signature scheme [7], we see that the use of the subgroup of order q does not significantly improve computational efficiency over the ElGamal signature scheme. Nevertheless, the use of the subgroup of the order q does achieve smaller signatures than those of the ElGamal signature scheme.

3.4 Digital signature schemes based on both the factoring and discrete logarithm problems

While the security of handwritten signature lies in the difficulty of producing undetectable forged signatures, the security of digital signature depends on the

intractability of mathematical hard problems. In the previous sections, we see that the security of RSA is based on the factoring problem, in the sense that if an attacker can solve the factoring problem, then he can achieve a total break of the system. On the other hand, in the last section we see a different approach for the hard problem: the security of the ElGamal signature scheme is based on the discrete logarithm problem.

In order to enhance security, several digital signature schemes based on both the factoring and discrete logarithm problems were proposed. In the following subsections, we gives a survey on these "hard" signature schemes and discuss their merits, demerits and security.

3.4.1 The Brickell-McCurley signature scheme

The Brickell-McCurley signature scheme [3] was the first digital signature scheme whose security is based on the difficulty of solving both the factoring and discrete logarithm problems.

The Brickell-McCurley signature scheme is a modification of the Schnorr signature scheme discussed in the last section. The merit of the Brickell-McCurley signature scheme is that while it can enhance security by basing on both the factoring and discrete logarithm problems, at the same time it retains nearly the same high level of efficiency as that of the Schnorr signature scheme.

Nevertheless, the Brickell-McCurley signature scheme is not exactly based on both the discrete logarithm and factoring problems. The reason is that in the Brickell-McCurley signature scheme, one of the hard problem, the factoring problem is to factor $(p-1)$, where p is a prime such that there exist integers q, w which are prime divisors of $(p-1)$. However, in the Brickell-McCurley

signature scheme, it is publicly known that the public key of a user contains an element α whose order is exactly equal to the unknown factor q of $(p-1)$. In other words, in the Brickell-McCurley signature scheme, α, p are public, moreover, it is publicly known that $\alpha^q \equiv 1 \bmod p$ and $q|p$. Therefore, the Brickell-McCurley signature scheme is not exactly based on both the discrete logarithm and factoring problems.

3.4.2 The Okamoto signature scheme

Okamoto [4] proposed another digital signature scheme whose security is based on both the factoring and discrete logarithm problems. A remarkable merit of the Okamoto signature scheme is that it is provably secure. However, again, the security of the Okamoto signature scheme is not exactly based on both the discrete logarithm and factoring problems. Instead, the Okamoto signature scheme is proved to be based on both the discrete logarithm problem and the finding order problem only.

In the Okamoto signature scheme, the finding order problem is defined as: *Given a prime p and a base element g, compute the order of g in Z_p^**. Thus, if one can solve the factoring problem, then he can solve the finding order problem, since the finding order problem can be solved by factoring $(p-1)$. However, on the other hand, if one can solve the finding order problem, then it does not mean that he can solve the factoring problem. Therefore, the security of the Okamoto signature scheme is not exactly based on both the discrete logarithm and factoring problems, since the Okamoto signature scheme is proved to be based on both the discrete logarithm problem and the finding order problem only.

Finally, concerning efficiency, the Okamoto signature scheme is less efficient than the Brickell-McCurley signature scheme in that it requires more computation and a larger signature size than does the Brickell-McCurley signature scheme.

3.4.3 The Harn signature scheme

In [8], Harn proposed a digital signature scheme whose security is claimed to be based on both the factoring and discrete logarithm problems. The Harn signature scheme has a merit that the modulus p associated with discrete logarithm and the modulus n associated with factoring are of nearly the same size, which allows the use of similar size moduli for same security level of the discrete logarithm and factoring problems.

However, in [19], Lee and Hwang show that in the Harn signature scheme, even if an adversary can solve the discrete logarithm only but cannot solve the factoring problem, the adversary still can forge signature in the Harn signature scheme with high probability. Therefore, the security of the Harn signature scheme is not based on both the discrete logarithm and factoring problems.

3.4.4 The Shao signature scheme

In [9], Shao proposed two digital signature schemes, both of which are claimed to have their security based on both the discrete logarithm and the factoring problems.

However, again, in [20], Li and Xiao show that the security of the Shao

signature schemes are not secure and are not based on any hard problem. Without the ability to solve both the discrete logarithm and factoring problems, an adversary still can forge signature in the Shao signature schemes.

3.4.5 The W. He signature scheme

Recently, W. He [5] proposed a new digital signature scheme, the security of which is claimed to be equivalent to computing both the discrete logarithm and the factoring problems.

There are two remarkable merits in the W. He signature scheme: First, users in the system can share a common modulus. Second, a user in the system needs to store only one number for her public key and only one number for her private key. As a result, the W. He signature scheme has an improvement in storage efficiency over the Brickell-McCurley signature scheme and the Okamoto signature scheme.

The W. He signature scheme is efficient in storage and computation, however, the security of the W. He signature scheme is not based on both the discrete logarithm and factoring problems, as proved in this thesis in the next chapter.

Chapter 4

Cryptanalysis of the digital signature scheme of W. He

To enhance security, W. He [5] recently proposed a new digital signature scheme, the security of which is claimed to be based on the difficulties of simultaneously solving the factoring and the discrete logarithm problems.

The digital signature scheme of W. He has a remarkable merit : not only does it permit users in the system to share a common modulus, but it also allows a user in the system to store only one number for her public key and only one number for her private key. Thus, the W. He signature scheme has an improvement in storage efficiency over the Brickell-McCurley signature scheme and the Okamoto signature scheme discussed in the previous chapter.

However, in contrast to He's claim, in this chapter we present a cryptanalysis of the digital signature scheme of W. He, and prove that the digital signature scheme of W. He is not based on the difficulties of simultaneously solving the factoring and the discrete logarithm problems.

For a self-contained and clear presentation of our cryptanalysis of the digital signature scheme of W. He, this chapter consists of two sections: i) In the first section, we introduce the digital signature scheme of W. He [5], ii) In the second section, we present a cryptanalysis of the digital signature scheme of W. He, and prove that the security of He's digital signature scheme is not based on the difficulties of simultaneously solving the factoring and the discrete logarithm problems.

4.1 The Digital Signature Scheme of W. He

Recently, in [5], W. He proposed a new digital signature scheme, the security of which is claimed to be based on the difficulties of simultaneously solving the factoring and the discrete logarithm problems in order to enhance security. This section, which introduces the digital signature scheme of W. He, serves as a preliminary for the cryptanalysis presented in the next section.

4.1.1 System setup in the W. He Digital Signature Scheme

In the system setup, a trusted centre selects a large prime $P = 4p_1q_1 + 1$, and an element g with order p_1q_1 in Z_P, where $p_1 = 2p_2+1$, $q_1 = 2q_2+1$ and p_1, q_1, p_2, q_2 are all distinct primes. A one-way hash function $f(\cdot)$ and the system parameters P, g are made public while p_1, q_1, p_2, q_2 are all kept secret and discarded.

Remark: Since P is public, $p_1q_1 = (P-1)/4$ is public.

4.1.2 Key generation in the W. He Digital Signature Scheme

After the system setup, each user in the system randomly selects a private key $x \in Z_{p_1q_1}$ such that $\gcd((x+x^{-1})^2, p_1q_1) = 1$ and computes the corresponding public key $y = g^{(x+x^{-1})^2} \bmod P$. The public key y is published and the private key x is secretly stored.

4.1.3 Signature generation in the W. He Digital Signature Scheme

Having generated her private key x and public key y, a signer (Alice) can use her private key x to sign a message m. To generate a signature for message m, the signer performs the following steps: i) Randomly select an integer $t \in Z_{p_1q_1}$ such that $\gcd((t+t^{-1})^2, p_1q_1) = 1$, then compute $r_1 = g^{(t+t^{-1})^2} \bmod P$ and $r_2 = g^{(t+t^{-1})^{-2}} \bmod P$. ii) Find s such that $(x+x^{-1}) = s \cdot (t+t^{-1}) + f(r_1, r_2, m) \cdot (t+t^{-1})^{-1} \pmod{p_1q_1}$, where f is the published one-way hash function. iii) Send signature (r_1, r_2, s) to the verifier (Bob).

4.1.4 Signature verification in the W. He Digital Signature Scheme

Upon receiving the signature (r_1, r_2, s) for the message m with respect to the signer (Alice), using signer's public key y, the verifier (Bob) verifies the signature by checking the following congruent equality:

$$y \stackrel{?}{\equiv} r_1^{s^2} r_2^{f^2(r_1,r_2,m)} g^{2s \cdot f(r_1,r_2,m)} \bmod P \tag{4.1}$$

If the equality holds, then (r_1, r_2, s) is a valid signature of the signer Alice on the message m.

4.2 Cryptanalysis of the digital signature scheme of W. He

In [5], the security of the W. He signature scheme is claimed to be based on the difficulties of simultaneously solving the factoring and discrete logarithm problems. After reviewing the digital signature scheme of W. He in the previous section, in this section we present a cryptanalysis of the digital signature scheme of W. He, and prove that the security of digital signature scheme of He is not based on the difficulties of simultaneously solving the factoring and discrete logarithm problems.

We present our cryptanalysis result in two subsections. In the first subsection, we derive two theorems on the security of the digital signature scheme of W. He. Then, in the second subsection, based on the two theorems in the first subsection, we design two algorithms for signature forgery in the digital signature scheme of W. He.

4.2.1 Theorems on the security of the digital signature scheme of W. He

In this subsection, we derive two theorems on the security of the digital signature scheme of W. He, which prove that the security of the W. He signature scheme is not based on both the factoring and discrete logarithm problems.

Theorem 4.2.1 *Given a message m, system parameters (P, g, f), and user Alice's public key y. Let (a) α, β be randomly chosen integers such that β is a power of g, (b) $\gamma = \beta^{f^2(1+\alpha P,\beta,m)} \bmod P$, (c) $\lambda = g^{2\cdot f(1+\alpha P,\beta,m)} \bmod P$, (d) $\sigma = \gamma^{-1} \cdot y \bmod P$, (e) $\delta : \lambda^\delta \equiv \sigma \bmod P$, (f) $r_1 = 1 + \alpha P$, $r_2 = \beta$, and $s = \delta$. Then, (r_1, r_2, s) is a valid signature of Alice for message m in He's digital signature scheme.*

Proof: It is sufficient to show that $(r_1 = 1 + \alpha P, r_2 = \beta, s = \delta)$ satisfies the signature verification equation of the W. He signature scheme, i.e. $y \equiv r_1^{s^2} r_2^{f^2(r_1,r_2,m)} g^{2s\cdot f(r_1,r_2,m)} \pmod P$, in He's scheme :

$$
\begin{aligned}
& r_1^{s^2} r_2^{f^2(r_1,r_2,m)} g^{2s\cdot f(r_1,r_2,m)} \\
\equiv\ & (1+\alpha P)^{\delta^2} \beta^{f^2(1+\alpha P,\beta,m)} g^{2\delta\cdot f(1+\alpha P,\beta,m)} \\
\equiv\ & (1+\alpha P)^{\delta^2} \cdot \gamma \cdot (g^{2\cdot f(1+\alpha P,\beta,m)})^{\delta} \\
\equiv\ & (1+\alpha P)^{\delta^2} \cdot \gamma \cdot \lambda^{\delta} \\
\equiv\ & \gamma \cdot \lambda^{\delta} \\
\equiv\ & \gamma \cdot \sigma \\
\equiv\ & \gamma \cdot (\gamma^{-1} y) \\
\equiv\ & y \pmod P
\end{aligned}
$$

□

Theorem 4.2.2 *Given a message m, system parameters (P, g, f), and user Alice's public key y. Let (a) λ be an integer randomly chosen in $Z_{p_1q_1}$, (b) $\beta = y^{\lambda^2} \bmod P$, (c) $\alpha : f(\alpha, \beta, m) \equiv \lambda^{-1} \bmod p_1q_1$, (d) $\sigma : g^\sigma \equiv \alpha \bmod P$,*

(e) $\gamma = -2\lambda^{-1}\sigma^{-1} \bmod p_1q_1$, *(f)* $r_1 = \alpha \bmod P$, $r_2 = \beta$, *and* $s = \gamma$. *Then,* (r_1, r_2, s) *is a valid signature of Alice for message* m *in He's digital signature scheme.*

Proof: It is sufficient to show that $(r_1 = \alpha, r_2 = \beta, s = \gamma)$ satisfies the signature verification equation of the W. He signature scheme, i.e. $y \equiv r_1^{s^2} r_2^{f^2(r_1,r_2,m)} g^{2s\cdot f(r_1,r_2,m)} \pmod P$, in He's scheme:

$$
\begin{aligned}
& r_1^{s^2} r_2^{f^2(r_1,r_2,m)} g^{2s\cdot f(r_1,r_2,m)} \\
\equiv\ & \alpha^{\gamma^2} \beta^{f^2(\alpha,\beta,m)} g^{2\gamma\cdot f(\alpha,\beta,m)} \\
\equiv\ & g^{\sigma\gamma^2} y^{\lambda^2 f^2(\alpha,\beta,m)} g^{2f(\alpha,\beta,m)\cdot\gamma} \\
\equiv\ & g^{\sigma\gamma^2 + 2f(\alpha,\beta,m)\cdot\gamma} y^{\lambda^2\cdot f^2(\alpha,\beta,m)} \\
\equiv\ & g^{\sigma\gamma^2 + 2\lambda^{-1}\cdot\gamma} y^{\lambda^2\cdot(\lambda^{-1})^2} \\
\equiv\ & (g^{\sigma\gamma + 2\lambda^{-1}})^{\gamma} y \\
\equiv\ & (g^{\sigma(-2\lambda^{-1}\sigma^{-1}) + 2\lambda^{-1}})^{\gamma} y \\
\equiv\ & (g^{-2\lambda^{-1} + 2\lambda^{-1}})^{\gamma} y \\
\equiv\ & y \pmod P
\end{aligned}
$$

□

4.2.2 Signature Forgery in the digital signature scheme of W. He

In this subsection, based on the two theorems in the previous subsection, we design two algorithms for signature forgery in the digital signature scheme of W. He.

4.2.2.1 Signature Forgery Algorithm requiring discrete logarithm only

By using *Algorithm 4.2.1* stated below, *without solving any factoring problem*, in the W. He signature scheme [5] an adversary can impersonate any user and forge signature for any desired message, provided that he can solve the discrete logarithm problem.

Given only the public key y of a signer. Without the knowledge of any valid signature and the private key of the signer, an adversary proceeds as stated in *Algorithm 4.2.1* to forge a valid signature (r_1, r_2, s) for any desired message m:

Algorithm 4.2.1 (Signature Forgery requiring discrete logarithm only)

INPUT: m, P, g, f, y.

OUTPUT: r_1, r_2, s.

1. Arbitrarily choose two integers α, β such that $\beta \not\equiv 0 \bmod P$, then set $r_1 = 1 + \alpha P$ and $r_2 = \beta$.
2. Compute $\gamma = \beta^{f^2(1+\alpha P, \beta, m)} \bmod P$, $\lambda = g^{2 \cdot f(1+\alpha P, \beta, m)} \bmod P$ and $\sigma = \gamma^{-1} \cdot y \bmod P$.
3. Find δ such that $\lambda^{\delta} \equiv \sigma \bmod P$.
4. Set $s = \delta$.

Remarks:

i. The above steps do not require the adversary to solve any factoring problem.

ii. In step 3, it is assumed that the adversary can solve the discrete logarithm problem, i.e. given values λ, σ, P, the adversary can find δ within polynomial time such that $\lambda^{\delta} \equiv \sigma \bmod P$.

The correctness of *Algorithm 4.2.1* is proved by *Theorem 4.2.1.*

4.2.2.2 Signature Forgery Algorithm requiring only discrete logarithm and hash function inversion

By using *Algorithm 4.2.2* stated below, in the W. He signature scheme [5], *without solving any factoring problem*, an adversary can impersonate any user and forge signature for any desired message, provided that he can solve the discrete logarithm problem and invert a hash function.

Given only the public key y of a signer. Without the knowledge of any valid signature and the private key of the signer, an adversary proceeds as stated in *Algorithm 4.2.2* to forge a valid signature (r_1, r_2, s) for any desired message m:

Algorithm 4.2.2 (Signature Forgery requiring only discrete logarithm and hash function inversion)

INPUT: m, P, g, f, y.

OUTPUT: r_1, r_2, s.

1. Compute $p_1q_1 = (P-1)/4$
2. Randomly choose an integer λ in $Z^*_{p_1q_1}$. Next, compute $\beta = y^{\lambda^2} \bmod P$.
3. Find α such that $f(\alpha, \beta, m) \equiv \lambda^{-1} \bmod p_1q_1$.
4. Find σ such that $g^\sigma \equiv \alpha \bmod P$.
5. Compute $\gamma = -2 \cdot \lambda^{-1} \cdot \sigma^{-1} \bmod p_1q_1$.
6. Compute $r_1 = \alpha \bmod P$, $r_2 = \beta$, $s = \gamma$.

Remarks:

i. The above steps do not require the adversary to solve any factoring problem.

ii. In step 3, it is assumed that the adversary is able to "invert" the hash function $f(.)$, i.e. given $\beta, m, \lambda^{-1}, p_1 q_1$, the adversary is able to find α within polynomial time such that $f(\alpha, \beta, m) \equiv \lambda^{-1} \bmod p_1 q_1$. In addition, if the adversary feels that it is difficult to invert a particular value of λ^{-1}, then he can choose another value for λ and repeats until a successful inversion is achieved.

iii. In step 4, it is assumed that the adversary can solve the discrete logarithm problem, i.e. given values α, g, P, the adversary can find σ within polynomial time such that $g^{\sigma} \equiv \alpha \bmod P$.

The correctness of *Algorithm 4.2.2* is proved by *Theorem 4.2.2.*

4.2.3 Remedy

To counteract the *signature forgery Algorithm 4.2.1*, a remedy is to forbid the first entry of the signature from being equal to 1 (modulo P). However, there is no trivial way to obviate the *signature forgery Algorithm 4.2.2.*

Chapter 5

Conclusions

While handwritten signature plays an important role in traditional commerce, digital signature plays an important role in electronic commerce. Signature forgery is of great concern for all signature schemes, regardless of whether they are paper signature schemes or digital signature schemes. Therefore, it is desirable to enhance the security of digital signature schemes, by using multiple (instead of a single) *"hard"* problems.

After introducing the necessary definitions and theorems on number theory and cryptology in chapter 2, chapter 3 presents a survey on remarkable digital signature schemes based on a single hard problem as well as those based on multiple hard problems to enhance security. In chapter 3, we have discussed the merits, demerits and security of a variety of remarkable digital signature schemes related to our cryptanalysis result presented in this thesis, including the RSA signature scheme whose security is based on factoring, the ElGamal signature scheme and Schnorr signature scheme whose security is based on discrete logarithm, the Brickell-McCurley signature scheme and Okamoto signature scheme

whose security is based on both the factoring and discrete logarithm problems, and the Harn signature scheme, Shao signature scheme, W. He signature scheme, each of which claims that its security is based on both the factoring and discrete logarithm problems.

In chapter 4 we have presented our cryptanalysis of the digital signature scheme of W. He. The first section of chapter 4 introduces the W. He signature scheme whose security is claimed to be based on both factoring and discrete logarithm, and which has a remarkable merit that not only does it permit users in the system to share a common modulus, but it also allows a user in the system to store only one number for her public key and only one number for her private key. Thus, the W. He signature scheme has an improvement in storage efficiency over the Brickell-McCurley signature scheme and the Okamoto signature scheme. Then, in the second part of chapter 4, we have derived a cryptanalysis of the W. He digital signature scheme and proved that the security of the W. He digital signature scheme is not based on both the factoring and the discrete logarithm problems. Signature forgery algorithms that do not require solving the factoring problem are also derived in chapter 4. Finally, a remedy is proposed to counteract one of the proposed signature forgery algorithms.

Bibliography

[1] Diffie, W. and Hellman, M. "New Directions in Cryptography." *IEEE Trans. on Information Theory*, IT-22, 1976, pp 644-654.

[2] Rivest, R., Shamir, A. , and Adleman, L. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." *Communications of the ACM*, 21(2), 1978, pp 120-126.

[3] Brickell, E., and McCurley, K. "An Interactive Identification Scheme Based on Discrete Logarithms and Factoring." *Journal of Cryptology*, 5(1), 1992, pp 29-39.

[4] Okamoto, T. "Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes." *Advances in cryptology — CRYPTO '92*, 1993, pp 31-53.

[5] He, W. "Digital signature scheme based on factoring and discrete logarithms." *Electronics Letters*, 37(4), 2001, pp 220-222.

[6] ElGamal, T. "A Public Key Cryptosystem and a Signature Scheme based on discrete logarithms." *IEEE Trans. on Information Theory*, IT-31, 1985, pp 469-472.

[7] Schnorr, C.P. "Efficient Signature Generation by Smart Cards." *Journal of Cryptology*, 4(3), 1991, pp 161-174.

[8] Harn, L. "Public-key cryptosystem design based on factoring and discrete logarithms." *IEE Proceedings - Computers and Digital Techniques*, 141(3), 1994, pp 193-195.

[9] Shao, Z. "Signature schemes based on factoring and discrete logarithms." *IEE Proceedings - Computers and Digital Techniques*, 145(1), 1998, pp 33-36.

[10] Ireland, K., and Rosen, M. "A classical introduction to modern number theory." *Springer-Verlag, 2nd ed.*, 1990.

[11] Schneier, B. "Applied Cryptography: Protocols, Algorithms, and Source Code in C." *John Wiley, 2nd ed.*, 1996, pp 256.

[12] Schnorr, C.P. "An Efficient Cryptographic Hash Function." *presented at rump session of CRYPTO 1991*, August 1991.

[13] Daemen, J., Bosselaers, A., Gouvaerts, R., and Vandewalle, J. "Collisions for Schnorr's Hash Function FFT-Hash Presented at Crypto '91." *Advances in cryptology — ASIACRYPT '91*, 1993, pp 477-480.

[14] Baritaud, T., Gilbert, H., and Girault, M. "FFT Hashing is not Collision-free." *Advances in cryptology — EUROCRYPT '92*, 1993, pp 35-44.

[15] Schnorr, C.P. "FFT-Hash II, Efficient Cryptographic Hashing." *Advances in cryptology — EUROCRYPT '92*, 1993, pp 45-54.

[16] Vaudenay, S. "FFT-Hash-II is not yet Collision-free." *Advances in cryptology — CRYPTO '92*, 1993, pp 587-593.

[17] Tillich, J., and Zemor, G. "Hashing with SL_2." *Advances in cryptology — CRYPTO '94*, 1994, pp 40-49.

[18] Charnes, C., and Pieprzyk, J. "Attacking the SL2 Hashing Scheme." *Advances in cryptology — ASIACRYPT '94*, 1995, pp 322-330.

[19] Lee, N., and Hwang, T. "Modified Harn signature scheme based on factorising and discrete logarithms." *IEE Proceedings - Computers and Digital Techniques*, 143(3), 1996, pp 196-198.

[20] Li, J., and Xiao, G. "Remarks on new signature scheme based on two hard problems." *Electronics Letters*, 34(25), 1998, pp 2401.