Practical Approaches to Attaining Security against Adaptively Chosen Ciphertext Attacks (Extended Abstract)

Yuliang Zheng and Jennifer Seberry *

The Centre for Computer Security Research
Department of Computer Science
University of Wollongong
Locked Bag 8844, Wollongong, NSW 2521
AUSTRALIA
E-mail: {yuliang,jennie}@cs.uow.edu.au

Abstract. This paper presents three methods for strengthening public key cryptosystems in such a way that they become secure against *adaptively* chosen ciphertext attacks. In an adaptively chosen ciphertext attack, an attacker can query the deciphering algorithm with any ciphertexts, *except* for the exact object ciphertext to be cryptanalyzed. The first strengthening method is based on the use of one-way hash functions, the second on the use of universal hash functions and the third on the use of digital signature schemes. Each method is illustrated by an example of a public key cryptosystem based on the intractability of computing discrete logarithms in finite fields. Two other issues, namely applications of the methods to public key cryptosystems based on other intractable problems and enhancement of information authentication capability to the cryptosystems, are also discussed.

1 Introduction

A considerable amount of research has been done in recent years, both from the theoretical [BFM88, NY90, DDN91, RS92] and practical [Dam92] points of view, in the pursuit of the construction of public key cryptosystems secure against chosen ciphertext attacks. In such an attack, the attacker (cryptanalyst) has access to the deciphering algorithm of a cryptosystem. The attacker can query the deciphering algorithm with any ciphertexts, obtain the matching plaintexts and use the attained knowledge in the cryptanalysis of an object ciphertext.

The theoretical results are appealing in that the schemes which embody them are provably secure under certain assumptions. However, most of these schemes are impractical due to the large expansion of the resulting ciphertext. The recent and notable schemes by Damgård overcome the problem of impracticality, but they are totally insecure against *adaptively* chosen ciphertext attacks in which

* Supported in part by the Australian Research Council under the reference numbers A49030136, A49130102 and A49131885.

an attacker has access to the deciphering algorithm even after he or she is given an object ciphertext to be cryptanalyzed. The attacker is allowed to query the deciphering algorithm with any ciphertext, *except* for the exact object ciphertext.

Adaptively chosen ciphertext attacks would impose serious problems on many services provided by modern information technology. To illustrate the possible attacks, consider the case of a security-enhanced electronic mail system where a public key cryptosystem is used to encipher messages passed among users. Nowadays it is common practice for an electronic mail user to include the original message he or she received into a reply to the message. For instance, a reply to a message may be as follows

```
  (original message)
> ......
> Hi, is Yum-Cha still on tonight ?
> ......

  (reply to the message)
......
Yes, it's still on.  I've already made the bookings.
......
```

this practice provides an avenue for chosen ciphertext attacks, as an attacker can send a ciphertext to a target user and expect the user to send back the corresponding plaintext as part of the reply. Now suppose that a user Alice is in the process of negotiating, through the electronic mail system, with two other users Bob and Cathy who are rivals of each other in a business. Let c be a ciphertext from Bob to Alice. Naturally, Cathy would like to know the contents of the communications between Alice and Bob. Cathy can obtain the ciphertext c by eavesdropping. However, it would be infeasible for her to extract its contents immediately. Instead, Cathy might try to discover *implicitly* the contents of c through discussions with Alice using the electronic mail. The problem facing Cathy is that she can not simply pass c to Alice with the hope that Alice would include the contents of c into her reply, as Alice would detect that c is actually a ciphertext created by Bob but not by Cathy. Nevertheless, if the cryptosystem is insecure against adaptively chosen ciphertext attacks, Cathy might still be able to obtain *indirectly* what she wants in the following way

1. Send Alice ciphertexts $c_1, c_2, \ldots, c_n$, none of which is the same as the object ciphertext c.
2. Receive the matching plaintext messages (hopefully) and
3. Extract the contents of c by the use of information obtained from the n plaintext-ciphertext pairs.

In this paper we present three pragmatic methods for immunizing public key cryptosystems against adaptively chosen ciphertext attacks. The first method is based on the use of one-way hash functions, the second on the use of universal hash functions and the third on the use of digital signature schemes. Each method is illustrated by an example of a public key cryptosystem based on the intractability of computing discrete logarithms in finite fields. Security of

the three cryptosystems against adaptively chosen ciphertext attacks is formally proved under reasonable assumptions.

In Section 2, we introduce notion and notations that are needed, and summarize various types of possible attack to cryptosystems. In Section 3 previous proposals together with their problems are reviewed. Our immunization methods are illustrated in Section 4, by three public key cryptosystems based on the intractability of computing discrete logarithms in finite fields. Section 5 is concerned with two other issues, namely applications of the immunization methods to public key cryptosystems based on other intractable problems, such as the problem of factoring large composite numbers, and the addition of information authentication capability to the three cryptosystems. Finally Section 6 presents some concluding remarks.

The reader is directed to [ZS93] where the three cryptosystems are formally proved to be secure against adaptively chosen ciphertext attacks.

2 Notion and Notations

We will be concerned with the alphabet $\Sigma = \{0,1\}$. The length of a string x over Σ is denoted by $|x|$, and the concatenation of two strings x and y is denoted by $x||y$. The bit-wise exclusive-or of two strings x and y of the same length is denoted by $x \oplus y$. The i-th bit of x is denoted by x_i and the substring of x from x_i to x_j, where $i \leqq j$, is denoted by $x_{[i...j]}$. $\#S$ indicates the number of elements in a set S, and $x \in_R S$ means choosing randomly and uniformly an element x from the set S. The Cartesian product of two sets S and T is denoted by $S \times T$.

Denote by $\mathbb{N}$ the set of all positive integers, and by n a security parameter which determines the length of messages, the length of ciphertexts, the security of cryptosystems etc. As in the Diffie-Hellman/ElGamal's public key scheme [DH76, ElG85], p is an n-bit prime and g is a generator for the multiplicative group $GF(p)^*$ of the finite field $GF(p)$. Both p and g are public. To guarantee the security of cryptosystems based on the discrete logarithm problem, the length n of p should be large enough, preferably $n > 512$, and $p-1$ should contain a large prime factor [PH78, LO91]. Unless otherwise specified, all exponentiation operations appearing in the remaining part of this paper are assumed to be over the underlying groups.

Note that there is a natural one-to-one correspondence between strings in Σ^n and elements in the finite field $GF(2^n)$. Similarly, there is a natural one-to-one correspondence between strings in Σ^n and integers in $[0, 2^n - 1]$. Therefore, we will not distinguish among strings in Σ^n, elements in $GF(2^n)$ and integers in $[0, 2^n - 1]$.

A *public key cryptosystem*, invented by Diffie and Hellman [DH76], consists of three polynomial time algorithms (C, E, D). C is called a *key-generation algorithm* which, on input n, generates probabilistically a pair (pk, sk) of public and secret keys. Following the tradition in the field, when a security parameter n is used as input to an algorithm, it will be represented by the all-1 string of n bits which is denoted by 1^n. E is called an *enciphering algorithm* which, on

input a public key pk and a plaintext message m, outputs a ciphertext c. Here m is chosen from a message space M_n. D is called a *deciphering algorithm* which, on input a secret key sk and a ciphertext c, outputs a message m or a special symbol $\emptyset$ meaning "no plaintext output". E and D satisfy the following unique decipherability condition, namely $D(sk, E(pk, m)) = m$.

There are four common types of attack to a cryptosystem, namely *ciphertext only attacks*, *known plaintext attacks*, *chosen plaintext attacks* and *chosen ciphertext attacks* [Riv90]. Related attacks against digital signatures are fully discussed in [GMR88].

In a ciphertext only attack, which is the least severe among the four types of attack, an attacker is given an object ciphertext and tries to find the plaintext which is hidden in the object ciphertext.

In a known plaintext attack, an attacker has a collection of plaintext-ciphertext pairs besides an object ciphertext. The attacker may use the knowledge gained from the pairs of plaintexts and ciphertexts in the cryptanalysis of the object ciphertext.

In a chosen plaintext attack, an attacker has access to the enciphering algorithm. During the cryptanalysis of an object ciphertext, the attacker can choose whatever plaintexts he or she desires, feed the enciphering algorithm with the desired plaintexts and obtain the corresponding ciphertexts. Note that this type of attack is always applicable to a public key cryptosystem, since the attacker always has access to the public enciphering algorithm.

In a chosen ciphertext attack, which is the most severe among the four types of attack, an attacker has access to the deciphering algorithm. The attacker can query the deciphering algorithm with any ciphertexts and obtain the corresponding plaintexts. Then the attacker can use the knowledge obtained in the query and answer process to extract the plaintext of an object ciphertext.

Researchers further distinguish two forms of chosen ciphertext attack: *indifferently chosen ciphertext attacks* and *adaptively chosen ciphertext attacks*. An indifferently chosen ciphertext attack is also called a *lunchtime attack* or a *midnight attack* [NY90]. In such an attack the ciphertexts fed into the deciphering algorithm are chosen without being related to the object ciphertext. However the ciphertexts fed into the deciphering algorithm may be correlated with one another. This form of attack models the situation where the attacker has access to the deciphering algorithm *before* he or she is actually given the object ciphertext.

In adaptively chosen ciphertext attacks all ciphertexts fed into the deciphering algorithm can be correlated to the object ciphertext. This form of attack is more severe than the indifferently chosen ciphertext attacks and it models the situation where the attacker has access to the deciphering algorithm even *after* he or she is given the object ciphertext. The attacker is thus permitted to give the deciphering algorithm any available ciphertexts, *except for* the exact object ciphertext, and obtain the matching plaintexts. See the Introduction for a practical application where adaptively chosen ciphertext attacks would be a considerable threat.

3 Problems with Previous Proposals

Rabin pioneered the research of constructing provably secure public key cryptosystems by designing a public key cryptosystem with the property that extracting the complete plaintext of an object ciphertext is computationally equivalent to factoring large numbers [Rab79]. Goldwasser and Micali invented the first public key cryptosystem that hides all partial information [GM84]. The cryptosystem is a probabilistic one and it enciphers a plaintext in a bit-by-bit manner. A common drawback of these and many other cryptosystems is that, although secure against chosen plaintext attacks, they are easily compromised by chosen ciphertext attackers. On the other hand, much progress has been made in recent years in the construction of public key cryptosystems secure against chosen ciphertext attacks. We will review this development, and point out problems and weakness of the proposed schemes.

3.1 Theoretical Results

Theoretical study into the construction of public key cryptosystems secure against chosen ciphertext attacks was initiated by Blum, Feldman and Micali [BFM88], who suggested the potential applicability of non-interactive zero-knowledge proofs to the subject. Naor and Yung carried further the study and gave the first concrete public key cryptosystem that is (semantically) secure against indifferently chosen ciphertext attacks [NY90]. Rackoff and Simon considered a more severe type of attack, namely adaptively chosen ciphertext attacks, and gave a concrete construction for public key cryptosystems withstanding the attacks [RS92]. In [DDN91] Dolev, Dwork and Naor proposed a non-malleable (against chosen plaintext attacks) public key cryptosystem and proved that the cryptosystem is also secure against adaptively chosen ciphertext attacks.

All of these cryptosystems are provably secure under certain assumptions. However since they rely heavily on non-interactive zero-knowledge proofs, the resulting ciphertexts are in general much longer than original plaintexts. This disadvantage makes the cryptosystems highly impractical and difficult to realize in practice.

3.2 Damgård's Schemes

In [Dam92], Damgård took a pragmatic approach to the subject. He proposed two simple public key cryptosystems that appear to be secure against indifferently chosen ciphertext attacks. The first is based on deterministic public key cryptosystems. Let (E_0, D_0) be the pair of enciphering and deciphering algorithms of a deterministic public key cryptosystem. Let (pk_1, sk_1) and (pk_2, sk_2) be two pairs of public and secret keys and h be an invertible one-to-one length-preserving function. The enciphering algorithm of Damgård's first cryptosystem operates in the following way:

$$E(pk_1, pk_2, m) = (E_0(pk_1, r),\ E_0(pk_2, h(r)) \oplus m) = (c_1, c_2)$$

where $m \in \Sigma^n$ is a plaintext message and $r \in_R \Sigma^n$ is a random string. The corresponding deciphering algorithm is as follows:

$$D(sk_1, pk_2, c_1, c_2) = E_0(pk_2, h(D_0(sk_1, c_1))) \oplus c_2$$

Damgård's second scheme is based on the Diffie-Hellman/ElGamal public key cryptosystem [DH76, ElG85], whose security relies on the intractability of computing discrete logarithms in finite fields. A user Alice's secret key is a pair (x_{A1}, x_{A2}) of elements chosen independently at random from $[1, p-1]$. Her public key is (y_{A1}, y_{A2}), where $y_{A1} = g^{x_{A1}}$ and $y_{A2} = g^{x_{A2}}$. When a user Bob wants to send an n-bit message m in secret to Alice, he sends her the following enciphered message

$$E(y_{A1}, y_{A2}, p, g, m) = (g^r,\ y_{A1}^r,\ y_{A2}^r \oplus m) = (c_1, c_2, c_3)$$

where $r \in_R [1, p-1]$. Note that here n is the length of the prime p. The deciphering algorithm for Alice, who possesses the secret key (x_{A1}, x_{A2}), is as follows

$$D(x_{A1}, x_{A2}, p, g, c_1, c_2, c_3) = \begin{cases} c_1^{x_{A2}} \oplus c_3 \text{ if } c_1^{x_{A1}} = c_2 \\ \\ \emptyset \qquad\qquad \text{otherwise} \end{cases}$$

Here $\emptyset$ is a special symbol meaning "no plaintext output".

Although Damgård's schemes are very simple and seem to be secure against indifferently chosen ciphertext attacks, they are *insecure* against adaptively chosen ciphertext attacks. Given an object ciphertext c ($c = (c_1, c_2)$ for the first scheme, and $c = (c_1, c_2, c_3)$ for the second scheme), an attacker can choose a random message m_r from Σ^n, calculate the bit-wise exclusive-or of m_r and the last part of the ciphertext c, and feed the deciphering algorithm with the modified ciphertext c'. The attacker will get $m' = m \oplus m_r$ as an answer, and obtain the desired message [2] m by computing $m' \oplus m_r$. Our cryptosystems to be described below share the same simplicity possessed by Damgård's cryptosystems, yet they attain a higher level of security, namely security against adaptively chosen ciphertext attacks.

4 Strengthening Public Key Cryptosystems

This section presents three simple methods for immunizing public key cryptosystems against chosen ciphertext attacks. The nature of the three immunization

[2] One might argue that, since at least half bits in the original ciphertext c remain untouched in the modified ciphertext c', adding a checking step to the deciphering algorithms would effectively thwart the attack. This countermeasure, however, does *not* work in general, as the deciphering algorithms may *not* know c. Even if the deciphering algorithms have a list of ciphertexts containing c, a more sophisticated attacker might still succeed in extracting m by generating c' in such a way that it passes the checking step.

methods is the same — they all immunize a public key cryptosystem by appending to each ciphertext a tag that is correlated to the message to be enciphered. This is also the main technical difference between our proposals and Damgård's schemes. The three methods differ in the ways in which tags are generated. In the first method tags are generated by the use of a one-way hash function, in the second method by the use of a function chosen from a universal class of hash functions, and in the third method by the use of a digital signature scheme. The second immunization method is superior to the other two immunization methods in that no one-way hash functions are needed. This property is particularly attractive given the current state of research, whereby many one-way hash functions exist, few are efficient, and even fewer are provably secure.

We will illustrate our immunization methods with cryptosystems based on the Diffie-Hellman/ElGamal public key scheme. In Section 5, applications of the immunization methods to cryptosystems based on other intractable problems will be discussed. Denote by G the cryptographically strong pseudo-random string generator based on the difficulty of computing discrete logarithms in finite fields [BM84, LW88, Per85]. G stretches an n-bit input string into an output string whose length can be an arbitrary polynomial in n. This generator produces $O(\log n)$ bits output at each exponentiation. In the authors' opinion, for practical applications the generator could produce more than $\frac{3n}{4}$ bits at each exponentiation, without sacrificing security. Recently Micali and Schnorr discovered a very efficient pseudo-random string generator based on polynomials in the finite field $GF(p)$ (see Section 4 of [MS91]). The generator can produce, for example, $\frac{n}{2}$ bits with 1.25 multiplications in $GF(p)$. The efficiency of our cryptosystems to be described below can be further improved if Micali and Schnorr's pseudo-random string generator is employed.

A user Alice's secret key is an element x_A chosen randomly from $[1, p-1]$, and her public key is $y_A = g^{x_A}$. It is assumed that all messages to be enciphered are chosen from the set Σ^P, where $P = P(n)$ is an arbitrary polynomial with $P(n) \geqq n$. Padding can be applied to messages whose lengths are less than n bits. In addition, let $\ell = \ell(n)$ be a polynomial which specifies the length of tags. It is recommended that ℓ should be at least 64 for the sake of security.

4.1 Immunizing with One-Way Hash Functions

Assume that h is a one-way hash function compressing input strings into ℓ-bit output strings. A user Bob can use the following enciphering algorithm to send in secret a P-bit message m to Alice.

Algorithm 1 $E_{owh}(y_A, p, g, m)$

1. $x \in_R [1, p-1]$.
2. $z = G(y_A^x)_{[1 \cdots (P+\ell)]}$.
3. $t = h(m)$.
4. $c_1 = g^x$.
5. $c_2 = z \oplus (m||t)$.
6. output (c_1, c_2).

end

The deciphering algorithm for Alice, who possesses the secret key x_A, is as follows:

Algorithm 2 $D_{owh}(x_A, p, g, c_1, c_2)$

1. $z' = G(c_1^{x_A})_{[1\cdots(P+\ell)]}$.
2. $w = z' \oplus c_2$.
3. $m' = w_{[1\cdots P]}$.
4. $t' = w_{[(P+1)\cdots(P+\ell)]}$.
5. if $h(m') = t'$ then
   output (m')
   else
   output $(\emptyset)$.

end

When messages are of n bits, i.e. $P = n$, instead of the one-way hash function h the exponentiation function can be used to generate the tag t. In this case, the enciphering algorithm can be modified as follows: (a) Change the step 2 to "$z = G(y_A^x)_{[1\cdots 2n]}$." (b) Change the step 3 to "$t = g^m$." The deciphering algorithm can be modified accordingly.

4.2 Immunizing with Universal Hash Functions

A class H of functions from Σ^P to Σ^ℓ is called a *(strongly) universal class of hash functions* [CW79, WC81] mapping P-bit input into ℓ-bit output strings if for every $x_1 \neq x_2 \in \Sigma^P$ and every $y_1, y_2 \in \Sigma^\ell$, the number of functions in H taking x_1 to y_1 and x_2 to y_2 is $\#H/2^{2\ell}$. An equivalent definition is that when h is chosen uniformly at random from H, the concatenation of the two strings $h(x_1)$ and $h(x_2)$ is distributed randomly and uniformly over the Cartesian product $\Sigma^\ell \times \Sigma^\ell$. Wegman and Carter found a nice application of universal classes of hash functions to unconditionally secure authentication codes [WC81].

Now assume that H is a universal class of hash functions which map P-bit input into ℓ-bit output strings. Also assume that $Q = Q(n)$ is a polynomial and that each function in H is specified by a string of exactly Q bits. Denote by h_s the function in H that is specified by a string $s \in \Sigma^Q$. The enciphering algorithm for Bob who wants to send in secret a P-bit message m to Alice is the following:

Algorithm 3 $E_{uhf}(y_A, p, g, m)$

1. $x \in_R [1, p-1]$.
2. $r = y_A^x$.
3. $z = G(r)_{[1\cdots P]}$.
4. $s = G(r)_{[(P+1)\cdots(P+Q)]}$.
5. $c_1 = g^x$.
6. $c_2 = h_s(m)$.
7. $c_3 = z \oplus m$.
8. output (c_1, c_2, c_3).

end

The deciphering algorithm for Alice, who possesses the secret key x_A, is as follows:

Algorithm 4 $D_{uhf}(x_A, p, g, c_1, c_2, c_3)$

1. $r' = c_1^{x_A}$.
2. $z' = G(r')_{[1\cdots P]}$.
3. $s' = G(r')_{[(P+1)\cdots(P+Q)]}$.
4. $m' = z' \oplus c_3$.
5. if $h_{s'}(m') = c_2$ then
   output (m')
   else
   output $(\emptyset)$.

end

Note that the second part $c_2 = h_s(m)$ in the ciphertext can be obscured in the same way as Algorithm 1. This would improve practical security of the cryptosystem, at the expense of more computation time spent in generating pseudo-random bits.

The following is a simple universal class of hash functions which is originated from linear congruential generators in finite fields. (See also Propositions 7 and 8 of [CW79].) Let k be an integer. For $k+1$ elements $a_1, a_2, \ldots, a_k, b \in GF(2^\ell)$, let s be their concatenation, i.e., $s = a_1||a_2||\cdots||a_k||b$, and let h_s be the function defined by $h_s(x_1, x_2, \ldots, x_k) = \sum_{i=1}^{k} a_i x_i + b$ where $x_1, x_2, \ldots, x_k$ are variables in $GF(2^\ell)$. Then the collection H of the functions h_s defined by all $k+1$ elements from $GF(2^\ell)$ is a universal class of hash functions. Functions in H compress $k\ell$-bit input into ℓ-bit output strings. By padding to input strings, these functions can be applied to input strings whose lengths are not exactly $k\ell$. In particular, when $k = \lceil \frac{P}{\ell} \rceil$, they can be used to compress P-bit input into ℓ-bit output strings. In this case, a function in H can be specified by a string of $Q = P + (1+\alpha)\ell$ bits, where $0 \leq \alpha = \frac{P \bmod \ell}{\ell} < 1$. This universal class of hash functions is particularly suited to the case where the length P of messages to be enciphered is much larger than the length ℓ of tags. We refer the reader to [WC81, Sti90] for other universal classes of hash functions.

4.3 Immunizing with Digital Signature Schemes

Assume that h is a one-way hash function compressing input strings into n-bit output strings. Also assume that Bob wants to send in secret a P-bit message m to Alice. The enciphering algorithm employed by Bob is the following:

Algorithm 5 $E_{sig}(y_A, p, g, m)$

1. $x \in_R [1, p-1]$.
2. $k \in_R [1, p-1]$ such that $\gcd(k, p-1) = 1$.
3. $r = y_A^{x+k}$.
4. $z = G(r)_{[1 \cdots P]}$.
5. $c_1 = g^x$.
6. $c_2 = g^k$.
7. $c_3 = (h(m) - xr)/k \bmod (p-1)$.
8. $c_4 = z \oplus m$.
9. output (c_1, c_2, c_3, c_4).

end

The corresponding deciphering algorithm for Alice, who possesses the secret key x_A, is as follows:

Algorithm 6 $D_{sig}(x_A, p, g, c_1, c_2, c_3, c_4)$

1. $r' = (c_1 c_2)^{x_A}$.
2. $z' = G(r')_{[1 \cdots P]}$.
3. $m' = z' \oplus c_4$.
4. if $g^{h(m')} = c_1^{r'} c_2^{c_3}$ then
   output (m')
   else
   output $(\emptyset)$.

end

Similar to the cryptosystem based on the use of universal hash functions described in Section 4.2, security of the cryptosystem can also be improved by hiding the third part $c_3 = (h(m) - xr)/k \bmod (p-1)$ with extra pseudo-random bits produced by the pseudo-random string generator G. In addition, when messages to be enciphered are of n bits, neither the one-way hash function h nor the pseudo-random string generator G is necessary. The enciphering algorithm for this case can be simplified by changing the step 4 of the above enciphering algorithm to "$z = r$." and the step 7 into "$c_3 = (m - xr)/k \bmod (p-1)$." The deciphering algorithm can be simplified accordingly.

The first three parts (c_1, c_2, c_3) of the ciphertext represents an adaptation of the ElGamal's digital signature. However, since everyone can generate these parts, they do *not* really form the digital signature of m. This immunization method was first proposed in [ZHS91], where other ways for generating the third part c_3 in the ciphertext were also suggested.

In [ZS93] it is proved that, under reasonable assumptions, all the three cryptosystems are secure against adaptively chosen ciphertext attacks. We introduce in the paper an interesting notion called *sole-samplability*, and apply the notion in the proofs of security.

5 Extensions of the Cryptosystems

We have focused our attention on cryptosystems based on the discrete logarithm problem in finite fields. The cryptosystems can also be based on discrete

logarithms over other kinds of finite abelian groups, such as those on elliptic or hyper-elliptic curves defined over finite fields [Kob87, Kob89]. Another variant of the cryptosystems is to have a different large prime for each user. This variant can greatly improve practical security of the cryptosystems when a large number of users are involved.

Our first two methods for immunization, namely immunization with one-way hash functions and immunization with universal hash functions, can be applied to public key cryptosystems based on other intractable problems. For example, the methods can be used to immunize the probabilistic public key cryptosystem proposed in [BG85], which is based on the intractability of factoring large composite numbers. The methods might be extended further in such a way that allows us to construct from *any* trap-door one-way function a public key cryptosystem secure against adaptively chosen ciphertext attacks.

Authentication is another important aspect of information security. In many situations, the receiver of a message needs to be assured that the received message is truly originated from its sender and that it has not been tampered with during its transmission. Researchers have proposed many, unconditionally or computationally, secure methods for information authentication [Sim88]. We take the second cryptosystem which uses universal has functions as an example to show that our cryptosystems can be easily added with information authentication capability.

To do so, it is required that the sender Bob also has a pair (y_B, x_B) of public and secret keys. Information authentication is achieved by letting Bob's secret key x_B be involved in the creation of a ciphertext. More specifically, we change the step 2 of the enciphering Algorithm 3 to "$r = y_A^{x_B+x}$." and the step 1 of the corresponding deciphering Algorithm 4 to "$r' = (y_B c_1)^{x_A}$." Although ciphertexts from Alice to Bob are indistinguishable from those from Bob to Alice, it is infeasible for a user differing from Alice and Bob to create a "legal" ciphertext from Alice to Bob or from Bob to Alice. This property ensures information authentication capability of the cryptosystem. It is not hard to see that computing $g^{x_1(x_2+x_3)}$ from g^{x_1}, g^{x_2} and g^{x_3}, and computing $g^{x_1x_2}$ from g^{x_1} and g^{x_2}, are equally difficult. Therefore the authentication-enhanced cryptosystem is as secure as the original one.

The first cryptosystem which is based on the use of a one-way hash function can be enhanced with information authentication capability in a similar way. For the third cryptosystem, the capability can be added by simply replacing x, a random string chosen from $[1, p-1]$, with Bob's secret key x_B.

6 Conclusions

We have presented three methods for immunizing public key cryptosystems against chosen ciphertext attacks, among which the second immunization method based on the use of universal hash functions is particularly attractive in that no one-way hash functions are needed. Each immunization method is illustrated by an example of a public key cryptosystem based on the intractability of com-

puting discrete logarithms in finite fields. The generality of our immunization methods is shown by their applicability to public key cryptosystems based on other intractable problems, such as that of factoring large composite numbers. An enhancement of information authentication capability to the example cryptosystems has also been suggested.

Acknowledgments

We would like to thank Thomas Hardjono for fruitful discussions, and to anonymous referees for helpful comments.

References

[BFM88] M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge proof systems and applications. In *Proceedings of the 20-th Annual ACM Symposium on Theory of Computing*, pages 103–112, 1988.

[BG85] M. Blum and S. Goldwasser. An efficient probabilistic public key encryption scheme which hides all partial information. In G. R. Blakeley and D. Chaum, editors, *Advances in Cryptology - Proceedings of Crypto'84*, Lecture Notes in Computer Science, Vol. 196, pages 289–299. Springer-Verlag, 1985.

[BM84] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4):850–864, 1984.

[CW79] J. Carter and M. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18:143–154, 1979.

[Dam92] I. Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In J. Feigenbaum, editor, *Advances in Cryptology - Proceedings of Crypto'91*, Lecture Notes in Computer Science, Vol.576, pages 445–456. Springer-Verlag, 1992.

[DDN91] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *Proceedings of the 23-rd Annual ACM Symposium on Theory of Computing*, 1991.

[DH76] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):472–492, 1976.

[ElG85] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31(4):469–472, 1985.

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[GMR88] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptively chosen message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[Kob87] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48:203–209, 1987.

[Kob89] N. Koblitz. Hyperelliptic cryptosystems. *Journal of Cryptology*, 1(3):139–150, 1989.

[LO91] B. A. LaMacchia and A. M. Odlyzko. Computation of discrete logarithms in prime fields. *Designs, Codes and Cryptography*, 1:47–62, 1991.

[LW88] D. L. Long and A. Wigderson. The discrete logarithm hides $O(\log n)$ bits. *SIAM Journal on Computing*, 17(2):363–372, 1988.

[MS91] S. Micali and C. P. Schnorr. Efficient, perfect polynomial random number generators. *Journal of Cryptology*, 3(3):157–172, 1991.

[NY90] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the 22-nd Annual ACM Symposium on Theory of Computing*, pages 427–437, 1990.

[Per85] R. Peralta. Simultaneous security of bits in the discrete log. In Franz Pichler, editor, *Advances in Cryptology - Proceedings of EuroCrypt'85*, Lecture Notes in Computer Science, Vol. 219, pages 62–72. Springer-Verlag, 1985.

[PH78] S. C. Pohlig and M. E. Hellman. An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. *IEEE Transactions on Information Theory*, IT-24(1):106–110, 1978.

[Rab79] M. Rabin. Digitalized signatures as intractable as factorization. Technical Report MIT/LCS/TR-212, MIT, Laboratory for Computer Science, 1979.

[Riv90] R. Rivest. Cryptography. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume A, Algorithms and Complexity*, chapter 13, pages 717–755. The MIT Press, Cambridge, Massachusetts, 1990.

[RS92] C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen-ciphertext attacks. In J. Feigenbaum, editor, *Advances in Cryptology - Proceedings of Crypto'91*, Lecture Notes in Computer Science, Vol.576, pages 433–444. Springer-Verlag, 1992.

[Sim88] G. J. Simmons. A survey of information authentication. *Proceedings of IEEE*, 76:603–620, 1988.

[Sti90] D. R. Stinson. Combinatorial techniques for universal hashing. Report Series #127, Department of Computer Science, University of Nebraska, Lincoln, November 1990. (Also submitted to *Journal of Computer and System Sciences*).

[WC81] M. Wegman and J. Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22:265–279, 1981.

[ZHS91] Y. Zheng, T. Hardjono, and J. Seberry. A practical non-malleable public key cryptosystem. Technical Report CS91/28, Department of Computer Science, University College, University of New South Wales, 1991.

[ZS93] Y. Zheng and J. Seberry. Immunizing public key cryptosystems against chosen ciphertext attacks. *Special Issue on Secure Communications, IEEE Journal on Selected Areas on Communications*, 1993. (to appear).