Cryptanalysis on an Improved Version of ElGamal-Like Public-Key Encryption Scheme for Encrypting Large Messages

Ting-Yi Chang,
Department of Industrial Education and Technology
National Changhua University of Education
No.1, Jin-De Road,
Changhua City, Taiwan, R.O.C.
E-mail: tychang@cc.ncue.edu.tw

Min-Shiang Hwang,
Department of Computer Science & Information Engineering
Asia University
1500, Lioufeng Rd., Wufeng,
Taichung, Taiwan, R.O.C.
( Corresponding author: Prof. Hwang)
E-mail: mshwang@mail.cyut.edu.tw

Wei-Pang Yang
Department of Information Management
National Dong Hwa University
1, Sec. 2, Da Hsueh Rd., Shou-Feng, Hualien, Taiwan, R.O.C.
E-mail: wpyang@mail.ndhu.edu.tw

Abstract. Hwang et al. proposed an ElGamal-like scheme for encrypting large messages, which is more efficient than its predecessor in terms of computational complexity and the amount of data transformation. They declared that the resulting scheme is semantically secure against chosen-plaintext attacks under the assumptions that the decision Diffie-Hellman problem is intractable. Later, Wang et al. pointed out that the security level of Hwang et al.’s ElGamal-like scheme is not equivalent to the original ElGamal scheme and brings about the disadvantage of possible unsuccessful decryption. At the same time, they proposed an improvement on Hwang et al.’s ElGamal-like scheme to repair the weakness and reduce the probability of unsuccessful decryption. However, in this paper, we show that their improved scheme is still insecure against chosen-plaintext attacks whether the system is operated in the quadratic residue modulus or not. Furthermore, we propose a new ElGamal-like scheme to withstand the adaptive chosen-ciphertext attacks. The security of the proposed scheme is based solely on the decision Diffie-Hellman problem in the random oracle model.
1. Introduction

Two typical primitives of the trapdoor one-way function are RSA (Rivest et al., 1978) and ElGamal (ElGamal, 1985). They are used in many cryptographic applications (Chang, 2008, Chang, 2009, Chang, 2010, Chmielowiec, 2010, Hwang et al., 2003, Wang and Hu, 2010, Yang et al., 2003), i.e., encryption and signatures. The difference between ElGamal function (Lee et al., 2009, Shen et al., 2003) and RSA function (bao et al., 2006, Hwang et al., 2000) is that probabilistic, rather than deterministic. In a probabilistic trapdoor one-way function, when encrypting a plaintext $x$ twice, the probability that we regain the same ciphertext $y$ must be negligibly small. Previously, what we have to face is that a passive attacker could break a cryptosystem only in the all-or-nothing (one-wayness) sense. However, this security notation which only deals with the case of passive attackers is not strong enough. On the contrary, the attacker maybe more active rather than passive; that is, she has more powerful capabilities to modify a ciphertext or to calculate a plaintext in some unspecified ways. To capture the powerful attackers, the stronger security notations are necessary and will be introduced in the following section.

1.1. Security Notations

To enhance the security notation, many stronger notations have been proposed. Bellare et al. (Bellare et al., 1998) uses the pair goal (GOAL) and adversary models (atk) to define the security notations of PKE and describe the relations among them. The goals GOAL={IND, NM} are defined as follows.

- **Indistinguishability** (IND): given the challenge ciphertext $y$, the adversary has no ability to obtain any information about the plaintext $x$.
- **Non-malleability** (NM): given the challenge ciphertext $y$, the adversary has no ability to decrypt $y$ to get a different ciphertext $y'$ and output a meaningful relation to relate the corresponding plaintexts $x$ and $x'$.

The adversary models ATK={CPA, CCA1, CCA2} are defined as follows.

- **Chosen-Plaintext Attack** (CPA) (Goldwasser and Micali, 1984): the adversary is only given the public key and she can obtain any ciphertext from any plaintext chosen by her. In the PKEs, this attack cannot be
avoided. It is considered as a basic requirement for most provably secure PKE.

- **Chosen-Ciphertext Attack (CCA1)** (Naor and Yung, 1990): not only given the public key, but also the adversary has to access a decryption oracle before being given the challenge ciphertext. It has also been called a lunch-time or midnight attack.

- **Adaptive Chosen-Ciphertext Attack (CCA2)** (Rackoff and Simon, 1991): The adversary queries the decryption before and after being challenged; her only restriction here is that she may not feed the decryption oracle with the challenge ciphertext itself. It has also been called a small-hours attack.

The following (Bellare et al., 1998, Fujisaki et al., 2001) are the relations among those GOAL-ATK.

\[
\begin{array}{c}
\text{NM-CPA} \leftrightarrow \text{NM-CCA1} \leftrightarrow \text{NM-CCA2} \\
\downarrow \times \times \downarrow \rightarrow \downarrow \uparrow \\
\text{IND-CPA} \leftrightarrow \text{IND-CCA1} \leftrightarrow \text{IND-CCA2}
\end{array}
\]

For \( A, B \in \text{GOAL-ATK} \), “\( A \rightarrow B \)” denotes \( A \) implies \( B \), which means if a PKE is secure in the sense of \( A \), it is also secure in the sense of \( B \). “\( A \nrightarrow B \)” denotes \( A \) doesn’t imply \( B \), which means if a PKE is secure in the sense of \( A \), it is not always secure in the sense of \( B \).

### 1.2. Relative Works

Many various PKEs have been proposed. The security of most of the widely-used PKEs is based on number-theoretic problems such as factoring integers and finding discrete logarithms over some cyclic group. Aim at to be secure in the stronger notations is more important. The general methodology for formally provable security is to reduce an alleged attack on an encryption scheme to a solution of an intractable problem.

Tsiouns and Yung (Tsiounis and Yung, 1998) showed that the IND-CPA of the ElGamal PKE operated in the quadratic residue modulo \( p \) is actually equivalent to the Decision Diffie-Hellman (DDH) problem. At the same time, they also proposed an enhanced ElGamal PKE is secure in the IND-CCA2 sense under the Random Oracle (RO) model and the decision Diffie-Hellman assumption. The RO is assumed to be an ideally random function when proving the security and it is replaced by a practical random-like function such as one-way hash function.
(Bellare and Rogaway, 1993). On the other hand, Cramer and Shoup (Cramer and Shoup, 1998) proposed a new public-key PKE based on the ElGamal, which is the first practical IND-CCA2 secure only under decision Diffie-Hellman assumption and the universal one-way hash functions, i.e., in the standard model (without the use of RO).

Most schemes are specified, they cannot be adopted by other schemes. There are two major conversions to convert existed trap-door one-way permutations to achieve IND-CCA2. Bellare-Rogaway conversion (Bellare and Rogaway, 1994) faces on the deterministic trap-door one-way permutations such as RSA and a comment (Shoup, 2001) revealed a flaw in that proof. Later, Fujisaki et al. (Fujisaki et al., 2001) find a way to rescue Bellare-Rogaway conversion for the trap-door partial-domain one-way permutations. On the other hand, Fujisaki-Okamoto conversion faces on the probabilistic trap-door one-way functions such as ElGamal. Both conversions are under the RO model and trap-door one-way function assumption.

Table 1. Assumptions and security notations of some related schemes

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Assumptions</th>
<th>GOAL-ATK</th>
</tr>
</thead>
<tbody>
<tr>
<td>ElGamal in QR, (Tsiounis and Yung, 1998)</td>
<td>DDH problem</td>
<td>IND-CPA</td>
</tr>
<tr>
<td>Tsiouns-Yung (Tsiounis and Yung, 1998)</td>
<td>DDH problem, RO</td>
<td>IND-CCA2</td>
</tr>
<tr>
<td>Shoup-Gennaro (Shoup and Gennaro, 1998)</td>
<td>DDH problem, RO</td>
<td>IND-CCA2</td>
</tr>
<tr>
<td>Cramer-Shoup (Cramer and Shoup, 1998)</td>
<td>DDH problem, UOWHF</td>
<td>IND-CCA2</td>
</tr>
<tr>
<td>Pointcheval (Pointcheval, 1999)</td>
<td>DCR problem, DPDL problem, RO</td>
<td>IND-CCA2</td>
</tr>
<tr>
<td>Paillier-Pointcheval (Paillier and Pointcheval, 1999)</td>
<td>DCR problem, DPDL problem, RO</td>
<td>IND-CCA2</td>
</tr>
<tr>
<td>Hwang et al. (Hwang et al., 2002)</td>
<td>DDH problem</td>
<td>IND-CPA</td>
</tr>
<tr>
<td>Bellare-Rogaway (Bellare and Rogaway, 1994)</td>
<td>deterministic trap-door partial-domain one-way permutations, RO</td>
<td>IND-CCA2</td>
</tr>
<tr>
<td>Fujisaki-Okamoto (Fujisaki et al., 2001)</td>
<td>probabilistic trap-door one-way functions, RO</td>
<td>IND-CCA2</td>
</tr>
</tbody>
</table>

universal one-way hash function (UOWHF), dependent-RSA (DRSA) problem, decision composite residuosity (DCR) problem, decision partial discrete logarithm (DPDL) problem

Table 1 shows the different assumptions and GOAL-ATK among some related schemes. As we realize it is not practical to implement the security proof in the RO-based technique since this kind of proof is heuristic only. However, the RO model usually has better efficiency and is still a useful test-bed to prove the security.
For encrypting a lengthy plaintext space efficiently in the PKE, Hybrid Public-Key Encryption (HPKE) schemes are devised (Abe et al., 2005), composed by two parts. The PKE scheme is used for encrypting a symmetric key $K$ and then the message $x$ is encrypted by the symmetric key. It is easy to construct a CCA2-secure HPKE (Abe et al., 2005), where the PKE is CCA2-secure and a symmetric encryption is secure against the passive attack such as the ciphertext is produced by $x \oplus K$ where $K$ is one-time use.

Hwang et al. (Hwang et al., 2002) consider a situation in the original ElGamal. When the plaintext $x$ is larger than the modulus $p$, it should be divided into several pieces $x_1, x_2, \ldots, x_n$ and each $x_i$ (for $i = 1$ to $n$) is smaller than $p$. Then we would need $n$ times to apply ElGamal encryption to obtain $n$ ciphertexts $y_i$’s. According $n$ ciphertexts $y_i$’s, we also need to apply $n$ times ElGamal decryption. It has the same results as in the HPKE schemes to encrypt the enough length of symmetric key $K$. Of course, the HPKE can firstly encrypt a smaller $K$ and then apply a pseudo-random bit generator on $K$ to generate an enough length of one-time pad to conform to the length of $x$. When the receiver decrypts the ciphertext of $K$, she must also apply the pseudo-random bit generator to obtain the one-time pad.

To withstand the reduce the computational complexity and the amount of data transformation as compared to the ElGamal, they proposed an ElGamal-like PKE for encrypting large messages and declared that the resulting scheme is in the IND-CPA sense under decision Diffie-Hellman assumption. Unfortunately, Wang et al. (Wang et al., 2006) pointed out that the security level of Hwang et al.’s ElGamal-like PKE is not equivalent to the original ElGamal scheme and brings about the disadvantage of possible unsuccessful decryption. At the same time, they proposed an improved version of Hwang et al.’s ElGamal-like PKE to repair the weakness and reduce the probability of unsuccessful decryption.

Wang et al.’s improved version of ElGamal-like PKE can be used in the situation for HPKE, which can remove a pseudo-random bit generator of the receiver since the encryption of PKE can directly encrypt a lengthy $K$ efficiently. However, we will show that their scheme is insecure in the IND-CPA sense in this paper. That is, their improved ElGamal-like PKE can’t provide the same security level as in the original the original ElGamal PKE. We also proposed an ElGamal-like PKE to satisfy the IND-CCA2 sense, which provides higher security confidence than satisfying the IND-CPA sense in Hwang et al.’s and Wang et al.’s PKEs. The security is under the assumption of the DDH problem in the random oracle model.
1.3. Outline of the Paper
The remainder of our paper is organized as follows. In Section 2, we shall give some definitions about the security of encryption scheme, quadratic residues, and Legendre symbol. In Section 3, we first give a brief review of the ElGamal which is not operated in the quadratic residue modulo \( p \) (denoted as \( \text{QR}_p \)) and then show that scheme is insecure in the IND-CPA sense. In Section 4, we separately show that the Wang et al.’s improved version of ElGamal-like PKE is insecure in the IND-CPA sense in \( \text{QR}_p \) and not in \( \text{QR}_p \). In Section 5, a new ElGamal-like PKE is proposed to satisfy the IND-CCA2 sense and its security is proven under the assumption of the DDH problem in the random oracle model. Then, we compare the computational complexity of our PKE with that of ElGamal PKE for encrypting a large message. Finally, we shall present our discussion and conclusion in Section 6.

2. Definitions and Security Models
In this section, we give some definitions about encryption scheme security, quadratic residues, and Legendre symbol as follows.

**Definition 1.** A function \( \varepsilon(k) \) is negligible if for every positive polynomial \( P(k) \in \mathbb{Z}[X] \), there is \( k_0 \), such that for every \( k \geq k_0 \), \( \varepsilon(k) < 1/P(k) \).

**Definition 2.** Let \( A \) be a probabilistic algorithm and let \( A(x_1, x_2, \ldots; r) \) be the result of running \( A \) on input \( x_1, x_2, \ldots \) and coins \( r \). We let \( y \leftarrow A(x_1, x_2, \ldots) \) denote the experiment of choosing \( r \) at random and letting \( y \) be \( A(x_1, x_2, \ldots; r) \). If \( S \) is a finite set, let \( x \leftarrow S \) be the operation of choosing \( x \) at random and uniformly from \( S \). For probability spaces \( S, T, \ldots \), the notation \( \Pr[\{x_1 \leftarrow S; x_2 \leftarrow T; \ldots : p(x_1, x_2, \ldots)\}] \) denotes after the ordered execution of the algorithms \( x_1 \leftarrow S, x_2 \leftarrow T, \ldots \), the probability that predicate \( p(x_1, x_2, \ldots) \) is true.

**Definition 3.** Let a triple of algorithm \( \Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D}) \) be a probabilistic PKE.

- The key generation algorithm \( \mathcal{K} \), is a probabilistic algorithm which on input \( 1^k \), where \( k \) is the security parameter, outputs a pair \((pk, sk)\) of matching public and secret key.
- The encryption algorithm \( \mathcal{E} \), is a probabilistic algorithm which on input a plaintext \( x \) and public key \( pk \), outputs a ciphertext \( y \).
- The decryption algorithm \( \mathcal{D} \), is a deterministic algorithm which on input ciphertext \( y \) and the secret key \( sk \), outputs the plaintext \( x \).

Here, we only give the definition of IND-ATK. The following sections will show that the ElGamal which is not operated in \( \text{QR}_p \) is not secure in
the IND-CPA sense, and Wang et al.’s ElGamal-like PKE is not secure in the IND-CPA sense either, operated in \( QR_p \) or not.

**Definition 4.** Let \( n \in \mathbb{N} \) and \( x \in \mathbb{Z} \). We call that \( x \) is quadratic residue modulo \( n \) if there is an element \( y \in \mathbb{Z} \) with \( x = y^2 \mod n \). Otherwise, \( x \) is called a quadratic non-residue modulo \( n \). The subgroup of \( \mathbb{Z}_n^* \) which consists of the residue classes represented by a quadratic residue, is denoted by \( QR_n \). The complement of \( QR_n \) is denoted by \( QNR_n = \mathbb{Z}_n^*/QR_n \).

**Definition 5.** Let \( p \) be a prime \( > 2 \), and let \( x \in \mathbb{Z} \) be prime to \( p \). \( \left( \frac{x}{p} \right) := \begin{cases} +1, & \text{if } [x] \in QR_p, \\ -1, & \text{if } [x] \in QNR_p, \end{cases} \)

is called the Legendre symbol of \( x \mod p \).

**Definition 6.** Let \( A = (A_1, A_2) \) be a pair of probabilistic algorithms, say Adversary for \( \Pi = (K, E, D) \). For \( ATK\{\text{CPA, CCA1, CCA2} \} \) and \( k \in \mathbb{N} \), denote the success event of \( A \) for \( \Pi \) by

\[
\text{Succ}_{A,\Pi}^{\text{ATK}}(k) = [(pk, sk) \leftarrow K(1^k); (x_0, x_1, state) \leftarrow A_1^O(pk); b \leftarrow_R \{0, 1\}; y \leftarrow E_{pk}(x_b) : A_2^O(x_0, x_1, state, y) = b],
\]

where the first two components of a triple \((x_0, x_1, state)\) are the plaintexts with the same length \(|x_0| = |x_1|\), and the last is a state information (including the public key pk) and some information to preserve. Here, \( O_1(\cdot), O_2(\cdot) \) are defined as follows:
- If \( ATK=\text{CPA} \) then \( O_1(\cdot)=\text{null} \) and \( O_2(\cdot)=\text{null} \);
- If \( ATK=\text{CCA1} \) then \( O_1(\cdot) = D_{sk}(\cdot) \) and \( O_2(\cdot)=\text{null} \);
- If \( ATK=\text{CCA2} \) then \( O_1(\cdot) = D_{sk}(\cdot) \) and \( O_2(\cdot) = D_{sk}(\cdot) \).

We denote the advantage of \( A \) for \( \Pi \) as

\[
\text{Adv}_{A,\Pi}^{\text{ATK}}(k) = 2 \cdot \Pr[\text{Succ}_{A,\Pi}^{\text{ATK}}(k)] - 1.
\]

We say that \( \Pi \) is secure in the IND-ATK sense if for any adversary \( A \) being polynomial-time in \( k \), \( \text{Adv}_{A,\Pi}^{\text{ATK}}(k) \) is negligible in \( k \).

### 3. Analysis of ElGamal PKE Scheme

Though the ElGamal operated in the quadratic residue modulo \( p \) has been showed that is secure in the IND-CPA sense under the Diffie-Hellman assumption (Tsiounis and Yung, 1998). In order to state our results clearly and precisely in breaking...
the Wang et al.’s ElGamal-like PKE (Wang et al., 2006), we begin with a review of the ElGamal which is not operated in the quadratic residue modulo p and then show that is insecure against IND-CPA.

3.1. ElGamal PKE Scheme Let \( \Pi = (K, E, D) \) be the ElGamal PKE.
- Key generation algorithm \( K : (pk, sk) \leftarrow K(1^k) \), \( pk = (p, g, Y) \) and \( sk = s \), where \( Y = g^s \mod p \), \( |p| = k, s \in \mathbb{Z}_p^* \), and \( \# < g >= p \). Let \( G_p \) be a group of prime order \( p \) of the multiplicative group \( \mathbb{Z}_p^* \). Let \( G \) be a prime

- Encryption algorithm \( E : (y_1, y_2) = E_{pk}(x; r) = (g^r \mod p, x \cdot Y^r \mod p) \), where message \( x \in \{0, 1\}^k \) and \( r \leftarrow \{0, 1\}^k \).
- Decryption algorithm \( D : x = D_{sk}(y_1, y_2) = y_2 \cdot (y_1^s)^{-1} \mod p \).

3.2. Security Analysis We can see that \( g \) is a primitive root of \( G_p \) by employing the key generation algorithm \( K \) in Section 3.1. Below, we first give the following lemmas and then show that encryption scheme is not secure in the IND-CPA sense.

**Lemma 1.** Let \( p \) be a prime \( > 2 \) and \( g \) be a primitive root of \( \mathbb{Z}_p^* \). Let \( [x] \in \mathbb{Z}_p^* \). Then \( x \in \mathbb{Q}R_p \) if and only if \( x = g^a \mod p \) some even number \( a, 0 \leq a < p - 1 \).

**Lemma 2.** The Legendre symbol is multiplicative in \( x \)

\[
\left( \frac{xy}{p} \right) = \left( \frac{x}{p} \right) \left( \frac{y}{p} \right)
\]

It means \( [xy] \in \mathbb{Q}R_p \) if and only if either both \( [x], [y] \in \mathbb{Q}R_p \) or both \( [x], [y] \in \mathbb{Q}NR_p \).

**Theorem 1.** Let \( \Pi = (K, E, D) \) be the ElGamal described in Section 3.1. An adversary \( A \) is a \((t, \epsilon)\)-breaker for \( \Pi(1^k) \) in IND-CPA if \( \text{Adv}_{\Pi,H}^{\text{CPA}} (k) \geq \epsilon \) and \( A \) runs within at most running time \( t \), where

\[
\epsilon = 1 \text{ and } t \leq t_1 + 3 \cdot t_QR.
\]

**Proof 1.** We construct a breaking algorithm \( A = (A_1, A_2) \) for \( \Pi = (K, E, D) \) as follows.
Adversary: \( A_1(pk) \)

Obtain \( \{x_0, x_1\} \), where \( x_0 \in \text{QR}_p \) and \( x_1 \in \text{QNR}_p \)

Return \((x_0, x_1, \text{state})\)

End.

Encryption oracle: \( O_{en}(x_0, x_1, pk) \)

\( b \leftarrow R \{0, 1\} \)

\((y_1, y_2) = E_{pk}(x_b; r) = (g^r \mod p, x_b \cdot Y^r \mod p)\)

End.

Adversary: \( A_2(x_0, x_1, \text{state}, (y_1, y_2)) \)

Case 1: \( Y \in \text{QR}_p \) and \( y_1 \in \{\text{QR}_p, \text{QNR}_p\} \)

If \( y_2 \in \text{QR}_p \), then outputs 0

If \( y_2 \in \text{QNR}_p \), then outputs 1

Case 2: \( y_1 \in \text{QR}_p \) and \( Y \in \text{QNR}_p \)

If \( y_2 \in \text{QR}_p \), then outputs 0

If \( y_2 \in \text{QNR}_p \), then outputs 1

Case 3: \( Y \in \text{QNR}_p \), \( y_1 \in \text{QNR}_p \)

If \( y_2 \in \text{QNR}_p \), then outputs 0

If \( y_2 \in \text{QR}_p \), then outputs 1

End.

We now analyze the successful probability of adversary \( A = (A_1, A_2) \). We define the following events. \( E_1 \) be the event \((Y \in \text{QR}_p) \land (y_1 \in \{\text{QR}_p, \text{QNR}_p\})\), \( E_2 \) be the event \((y_1 \in \text{QR}_p) \land (Y \in \text{QNR}_p)\) and \( E_3 \) be the event \((Y \in \text{QNR}_p) \land (y_1 \in \text{QNR}_p)\).

Let \( b' \) be the output of \( A_2 \). For Case 1, \( Y = g^s \in \text{QR}_p \). By Lemma 1, \( s \) is even, no matter what \( y_1 \in \text{QR}_p \) or \( y_1 \in \text{QNR}_p \), we know that \( Y^r = g^{sr} \in \text{QR}_p \).

We see that \( A_2 \) will output the correct \( b' = 0 \) (\( b' = 1 \)) if and only if \( y_2 \in \text{QR}_p \) (\( y_2 \in \text{QNR}_p \)). This is due to the multiplicative property of Legendre symbol in Lemma 2 as follows.

\[
\left( \frac{y_2}{p} \right) = \left( \frac{x_b}{p} \right) \left( \frac{Y^r}{p} \right)
\]

Therefore, the condition probability \( \text{Pr}[b = b'|E_1] = 1 \) and the probability \( \text{Pr}[E_1] = 1/2 \). For the same reason, in Case 2, the condition probability \( \text{Pr}[b = b'|E_2] = 1 \).

Note that \((y_1 \in \text{QR}_p) \land (Y \in \text{QR}_p)\) is included in the event \( E_1 \) and the probability \( \text{Pr}[E_1] = 1/4 \). For Case 3, \( Y \in \text{QNR}_p \) and \( y_1 \in \text{QNR}_p \), by Lemma 1, \( s \) and \( r \) are odd, \( Y^r = g^{sr} \in \text{QNR}_p \). \( A_2 \) will output the correct \( b' = 0 \) (\( b' = 1 \)) if and only
if \( y_2 \in \text{QR}_p \) (\( y_2 \in \text{QNR}_p \)). Thus, the condition probability \( \Pr[b = b'|E_3] = 1 \) and the probability \( \Pr[E_3] = 1/4 \). By the law of total probability,

\[
\Pr[\text{Succ}^\text{CPA}_{A,\Pi}(k)] = \Pr[b = b'] = \sum_{i=1}^{3} \Pr[b = b'|E_i] \cdot \Pr[E_i] = 1 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} + 1 \cdot \frac{1}{4} = 1.
\]

we have \( \text{Adv}^\text{CPA}_{A,\Pi}(k) = 2 \cdot \Pr[\text{Succ}^\text{CPA}_{A,\Pi}(k)] - 1 = 1 \).

Thus, we have the ability to distinguish the distinct plaintexts \( x_0 \) and \( x_1 \). To secure against IND-CPA, for security parameter \( k \), primes \( p \) and \( q \) are chosen such that \( p = 2q + 1 \) (\( q \) is called a Sophie-Germain prime if \( p \) is also a prime), where \(|p| = k\) and \(|q| = k - 1\). Then a unique subgroup \( G_q \) of prime order \( q \) of the multiplicative group \( \mathbb{Z}_p^* \) and \( g \) of \( G_q \) are defined. In other words, the key generation \( K \) should be modified as \( \tilde{K} \).

- Key generation \( \tilde{K} : (pk, sk) \leftarrow \tilde{K}(1^k), pk = (p, g, Y) \) and \( sk = (p, g, s) \), where \( Y = g^s \mod p \), \(|p| = k\), \(|q| = k - 1\), \( g = h^2 \mod p \), \( s \in \mathbb{Z}/q\mathbb{Z} \), and \(|< g >| = q\).

Since \( g \) generates all the quadratic residues in \( \text{QR}_p \) and the message for encrypting is needed to be a QR, the algorithm \( A \) does not work.

A value is determined whether it is in \( \text{QR}_p \) or not can be computed efficiently by Euler’s criterion in a polynomial time. Let \( t_{QR} \) be the time of determining whether a value is in \( \text{QR}_p \) or not. Let \( t_1 \) be the time of choosing two messages \( x_0 \in \text{QR}_p \) and \( x_1 \in \text{QNR}_p \). Then, from the specification of \( A = (A_1, A_2) \), it runs within at most 3 times \( t_{QR} \) in Case 2 or Case 3. Hence, \( t \leq t_1 + 3 \cdot t_{QR} \) and it is in a polynomial time.

4. Analysis of ElGamal-Like PKE Scheme
The ElGamal should employ the key generation algorithm \( \tilde{K} \) to ensure that the IND-CPA sense. However, in this section, we will show that even if the ElGamal-like PKE are given the same repair, it is still insecure in the IND-CPA sense.

4.1. ElGamal-Like PKE Scheme
Let \( \Pi' = (K', E', D') \) be Wang et al.’s ElGamal-like PKE.

- Key generation algorithm \( K' : (pk, sk) \leftarrow K'(1^k), pk = (p, g, Y) \) and \( sk = (p, g, s) \), where \( Y = g^s \mod p \), \(|p| = k\), \( s \in \mathbb{Z}_p^* \), and \(|< g >| = p\).
Note that the prime $p$ should be chosen such that the smallest positive integer $T$ for $2^T + 1 = 2 \mod p - 1$ is as large as possible (Please see (Wang et al., 2006) for more details). Otherwise, $g^{2^j} \mod p$ cannot generate all the elements in $\mathbb{Z}_p^*$ where $j = 1, 2, \ldots, p - 1$.

- Encryption algorithm $\mathcal{E}'$:

$$(y_1, y_2, y_3) = \mathcal{E}_\text{pk}'(x_i; r_1; r_2) = \left(g^{r_1} \mod p, g^{r_2} \mod p, x_i \cdot (Y^{r_1} \oplus (Y^{r_2})^i) \mod p\right),$$

where message $x \in \{0, 1\}^k$, $x$ is divided into $x_1, x_2, \ldots, x_n$ ($|x_1| = |x_2| = \cdots = |x_{n-1}|$, $n = \lceil |x| / k \rceil$, $|x_n| = |x| \mod k$, and each $x_i < p$) and $r_1, r_2 \leftarrow_R \{0, 1\}^k$. The notation $\oplus$ denotes as the bit-wise exclusive-or operation.

- Decryption algorithm $\mathcal{D}'$:

$$x_i = \mathcal{D}_\text{sk}'(y_1, y_2, y_3, i) = y_{3, i} \cdot (y_1^i \oplus (y_2^i)^i)^{-1} \mod p,$$
$$x = x_1 x_2 \cdots x_n.$$  
This scheme is designed for encrypting large messages, which will more efficient than the ElGamal. Here, we consider the same situation in the original ElGamal where the message $x < p$ is for encrypting as follows.

- Encryption algorithm $\mathcal{E}''$:

$$(y_1, y_2, y_3) = \mathcal{E}_\text{pk}''(x; r_1; r_2) = \left(g^{r_1} \mod p, g^{r_2} \mod p, x \cdot (Y^{r_1} \oplus Y^{r_2}) \mod p\right),$$

where message $x \in \{0, 1\}^k$, $x < p$, and $r_1, r_2 \leftarrow_R \{0, 1\}^k$.

- Decryption algorithm $\mathcal{D}''$:

$$x = \mathcal{D}_\text{sk}''(y_1, y_2, y_3) = y_3 \cdot (y_1^i \oplus y_2^i)^{-1} \mod p.$$  

4.2. Security Analysis

In the following theorem, we prove that Wang et al.’s ElGamal-like PKE in Section 4.1 is insecure in the IND-CPA sense and has the probability to make that cryptosystem failed.

**Theorem 2.** Let $\Pi' = (\mathcal{K}'', \mathcal{E}'', \mathcal{D}'')$ be the ElGamal-like PKE described in Section 4.1. An adversary $\mathcal{A}'$ is a $(t', \epsilon')$-breaker for $\Pi'$ in IND-CPA if $\text{Adv}^{\text{CPA}}_{\Pi'}(k) \geq \epsilon'$ with the event $\text{Fail}$ does not occur, and $\mathcal{A}'$ runs within at most running time $t'$, where

$$\epsilon' = 1 \text{ and } t' \leq t_1 + 3 \cdot t_{\text{QR}}.$$
Proof 2. We give a simple example and then analyze the results as follows. In the key generation algorithm $K'$, for $p = 7$, we select a generator $g = 5$ of $\mathbb{Z}_p^*$. It satisfies the requirement of $p$ such that the smallest positive positive integer $T$ for $2^{T+1} \equiv 2 \pmod{p-1}$ is as large as possible. In this example, $2^j \mod p$ generates $\{5, 7\}$ for all integer $j$ in $[1, 6]$ since the smallest positive integer such that $2^{T+1} \equiv 2 \pmod{p-1}$ is 2. Consider the sets $QR_p = \{1, 2, 4\}$ and $QNR_p = \{3, 5, 6\}$. By Lemma 1, the following situations (mod $p$ is abridged) are considered.

**Situation 1:** $Y^{r_1} \in QR_p$ and $Y^{r_2} \in QR_p$

The values of computing $Y^{r_1} \oplus Y^{r_2}$ are in the set $S_1 = \{1 \oplus 1, 1 \oplus 2, 1 \oplus 4, 2 \oplus 1, 2 \oplus 2, 2 \oplus 4, 4 \oplus 1, 4 \oplus 2, 4 \oplus 4\} = \{0, 3, 5, 3, 0, 6, 5, 6, 0\}$.

**Situation 2:** $Y^{r_1} \in QR_p$ and $Y^{r_2} \in QNR_p$

The values of computing $Y^{r_1} \oplus Y^{r_2}$ are in the set $S_2 = \{1 \oplus 3, 1 \oplus 5, 1 \oplus 6, 2 \oplus 3, 2 \oplus 5, 2 \oplus 6, 4 \oplus 3, 4 \oplus 5, 4 \oplus 6\} = \{2, 4, 0, 1, 0, 4, 0, 1, 2\}$.

**Situation 3:** $Y^{r_1} \in QNR_p$ and $Y^{r_2} \in QNR_p$

The values of computing $Y^{r_1} \oplus Y^{r_2}$ are the same as in $S_2$. $S_2 = S_3 = \{2, 4, 0, 1, 0, 4, 0, 1, 2\}$.

**Situation 4:** $Y^{r_1} \in QNR_p$ and $Y^{r_2} \in QNR_p$

The values of computing $Y^{r_1} \oplus Y^{r_2}$ are in the set $S_4 = \{3 \oplus 3, 3 \oplus 5, 3 \oplus 6, 5 \oplus 3, 5 \oplus 5, 5 \oplus 6, 6 \oplus 3, 6 \oplus 5, 6 \oplus 6\} = \{0, 6, 5, 6, 0, 3, 5, 3, 0\}$.

We can see that the values of $Y^{r_1} \oplus Y^{r_2}$ has the probability to be 0, no matter what plaintext $x$ is input to encrypt algorithm $E'$, the value of $y_3 = x \cdot (Y^{r_1} \oplus Y^{r_2})$ is equal to 0. The encrypt algorithm $E'$ is failed, together with the decrypt algorithm $D'$. Obviously, the probability of $\Pi' = (K', E', D')$ crashed in the above situations is $1/3$, denoted as $Pr[\text{Fail}] = 1/3$.

If the encryption algorithm $E'$ chooses $r_1$ or $r_2 \leftarrow_R \{0, 1\}^k$ again to avoid the case $Y^{r_1} \oplus Y^{r_2} = 0$, we still can construct a breaking algorithm $A' = (A'_1, A'_2)$ in the IND-CPA sense for $\Pi' = (K', E', D')$.

**Adversary:**
\[
A'_1(pk) \quad \text{Obtain } \{x_0, x_1\}, \text{ where } x_0 \in QR_p \text{ and } x_1 \in QNR_p \\
\text{Return } (x_0, x_1, \text{state})
\]

**Encryption oracle:**
\[
\mathcal{O}_{en}(x_0, x_1, pk) \\
b \leftarrow_R \{0, 1\} \\
(y_1, y_2, y_3) = E'_p(x_0; r_1; r_2) = (g^{x_1}, g^{x_2}, x_0 \cdot (Y^{r_1} \oplus Y^{r_2}))
\]

**End.**
The successful probability of adversary $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ is similar to $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ if $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ is not crashed, i.e., $\text{Fail}$ does not occur. By the multiplicative property of Legendre symbol,

$$\left( \frac{y_3}{p} \right) = \left( \frac{x_h}{p} \right) \left( \frac{Y^{r_1} \oplus Y^{r_2}}{p} \right),$$

the conditional probability $\Pr[\text{Adv}_{\mathcal{A}', \Pi}^{\text{CPA}}(k)|\neg \text{Fail}]$ is equal to 1 and $\text{Adv}_{\mathcal{A}', \Pi'}^{\text{CPA}}(k) = 2 \cdot \Pr[\text{Succ}_{\mathcal{A}', \Pi'}^{\text{CPA}}(k)|\neg \text{Fail}] - 1 = 1$. For the same reason, from the specification of $\mathcal{A}'$, it runs within at most $t' \leq t_1 + 3 \cdot t_{QR}$.

If we attempt to repair this scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ in the same fashion as in Section 3.2, the key generation algorithm $\mathcal{K}'$ is replaced as $\tilde{\mathcal{K}}$, and then the PKE becomes $\Pi'' = (\tilde{\mathcal{K}}, \mathcal{E}', \mathcal{D}')$. The following theorem will show that $\Pi'' = (\tilde{\mathcal{K}}, \mathcal{E}', \mathcal{D}')$ is still insecure in the IND-CPA sense.

**Theorem 3.** Let $\Pi'' = (\tilde{\mathcal{K}}, \mathcal{E}', \mathcal{D}')$ be the ElGamal-like PKE operated in $\text{QR}_p$. An adversary $\mathcal{A}''$ is a $(t'', \epsilon'')$-breaker for $\Pi''$ in IND-CPA if $\text{Adv}_{\mathcal{A}'', \Pi''}^{\text{CPA}}(k) \geq \epsilon''$ with the event $\text{Fail}$ does not occur, and $\mathcal{A}''$ runs within at most running time $t''$, where

$$\epsilon'' = 1 \text{ and } t'' \leq t_1 + t_{QR}.$$
We also give an example for the key generation algorithm \( \tilde{K} \), where \( q = 3 \), \( p = 2q + 1 = 7 \), \( h = 5 \), \( g = h^2 \mod p = 4 \). Obviously, \( g \in QR_p \), therefore, the group is in \( QR_p \), where \( QR_p = \{1, 2, 4\} \). The value of \( Y^{r_1} \oplus Y^{r_2} \mod p \) are in the set \( S_1 \) as the same as in Situation 1 of Theorem 2. \( \Pi'' = (\tilde{K}, E', D') \) has the probability to fail as follows:

\[
\Pr[\text{Fail}] = \Pr[\text{Fail} | Y^{r_1} \in QR_p] \cdot \Pr[Y^{r_1} \in QR_p]
\]

\[
= \frac{3}{9} \cdot 1 = \frac{1}{3}.
\]

A breaking algorithm \( A'' = (A''_1, A''_2) \) in the IND-CPA sense for \( \Pi'' = (\tilde{K}, E', D') \) is as follows:

**Proof 3.**

**Adversary:** \( A''_1(pk) \)

Obtain \( \{x_0, x_1\} \), where \( x_0 \in QR_p \) and \( x_1 \in QNR_p \)

Return \( (x_0, x_1, state) \)

End.

**Encryption oracle:** \( \mathcal{O}_{en}(x_0, x_1, pk) \)

\( b \leftarrow \mathcal{U}\{0, 1\} \)

\( (y_1, y_2, y_3) = E'_{pk}(x_b; r_1; r_2) = (g^{r_1}, g^{r_2}, x_b \cdot (Y^{r_1} \oplus Y^{r_2})) \)

End.

**Adversary:** \( A''_2(x_0, x_1, state, (y_1, y_2, y_3)) \)

Case 1: If \( y_3 \in QR_p \), then outputs 1

Case 2: If \( y_3 \in QNR_p \), then outputs 0

End.

Except the values when \( Y^{r_1} \oplus Y^{r_2} = 0 \), the Legendre symbol of \( Y^{r_1} \oplus Y^{r_2} \) is

\[
\left( \frac{Y^{r_1} \oplus Y^{r_2}}{p} \right) = -1,
\]

By the multiplicative property of Legendre symbol,

\[
\left( \frac{y_3}{p} \right) = \left( \frac{x_b}{p} \right) \left( \frac{Y^{r_1} \oplus Y^{r_2}}{p} \right),
\]
we can determine \( x_b \) is \( x_0 \in \mathbb{QR}_p \) or \( x_1 \in \mathbb{QR}_p \), according to the Legendre symbol \( \left( \frac{y_a}{p} \right) \). This forms Case 1 and Case 2 of \( A''_2 \), respectively. The advantage of \( A'' \) for \( \Pi'' \) is \( \text{Adv}_{A''_{\Pi''}}(k) = 2 \cdot \text{Pr}[\text{Succ}_{A''_{\Pi''}}(k) = \text{Fail}] = 1 = 1 \).

From the specification of \( \Pi'' \), it runs within at most \( t'' \leq t_1 + t_{QR} \). Obviously, the both breaking algorithms \( A' = (A_1', A_2') \) and \( A'' = (A_1'', A_2'') \) are in a polynomial time in Theorem 4.2 and Theorem 4.3, respectively.

We can see that no matter what Wang et al.’s ElGamal-like PKE employs \( K' \) or \( \tilde{K} \), the scheme is insecure in the IND-CPA sense, even the cryptosystem will be failed to encrypt and/or decrypt. Though the probability of event \( \text{Fail} \) will decrease when we chose a large prime \( q \) or \( p \) (the security parameter \( k \)), for both \( \Pi' = (K', E', D') \) and \( \Pi'' = (\tilde{K}, E', D') \), the values after exclusive-or operation may not in the group \( G_p \) and \( G_q \), respectively. This results in their scheme is insecure in the IND-CPA sense.

5. The Proposed ElGamal-Like Encryption Scheme

In this section, an ElGamal-like PKE is proposed and then we show that the proposed ElGamal-like PKE satisfies the IND-CCA2 sense under the DDH problem in the random oracle model.

5.1. ElGamal-Like PKE Scheme

Let \( \Pi^\dagger = (K^\dagger, E^\dagger, D^\dagger) \) be the ElGamal-extended encryption scheme.

- Key generation algorithm \( K^\dagger \): \((pk, sk) \leftarrow K^\dagger(1^k)\), \( pk = (p, g, Y) \) and \( sk = s \), where \( Y = g^s \mod p \), \( p = 2q + 1 \), \( # < h > = p \), \( g = h^2 \mod p \), \( s \in \mathbb{Z}/q\mathbb{Z} \), and \( # < g > = q \). Let \( k = k_0 + 2k_1 + 1 \).
- Hash functions \( H \) and \( J \): \( H : \{0, 1\}^{k_0 + 2k_1} \rightarrow \{0, 1\}^t \), \( J : \{0, 1\}^k \rightarrow \{0, 1\}^k \).
- Encryption algorithm \( E^\dagger \):

\[
(y_1', y_2', y_{3,i}) = E^\dagger_{pk}(x_i; r_1; r_2),
\]

1. Concatenate \( X_i = x_i || r_1 || r_2 \), where \( x_i \in \{0, 1\}^{k_0} \), \( r_1, r_2 \in_R \{0, 1\}^{k_1} \in \mathbb{Z}_q \), and \( || \) denotes concatenation.
2. Compute \( J_i = J(Y^{r_1} \mod p) \).
3. Compute \( (y_1, y_{3,i}) = (g^{r_1} \mod p, (X_i || H(X_i)) \cdot Y^{r_2} \mod p) \).
4. Compute \( (y_1', y_2', y_{3,i}) = (y_1, g^{r_2} \mod p, y_{3,i} \cdot J_i \mod p) \).
16

Ting-Yi Chang et al.

- Decryption algorithm $D^1$:

$$x_i = D^1_{sk}(y'_1, y'_2, y'_3, i),$$

1. Compute $J_i = J(y'_2 \cdot r_2 \mod p)$.
2. Compute $W_i = (y'_3 \cdot J_i^{-1}) \cdot (y'_s)^{-1} \mod p$.
3. Output

$$\begin{cases} 
[W_i][a], & \text{if } H([W_i][a+2k_i]) = [W_i][b] \\
null, & \text{otherwise}
\end{cases}$$

The notations of $[W_i][a]$ and $[W_i][b]$ denote the first $a$-bit and the last $b$-bit of $W_i$, respectively. Finally, the whole plaintext $x$ can be concatenated as $x_1, x_2, \cdots, x_n$.

There is an additional random value $J_i$ for each $x_i$. Even if there are only two random numbers $r_1$ and $r_2$, the hash value $J_i$ still makes the encryption scheme probabilistic. If the adversary can obtain the hash value $J(Y_i \cdot r \mod p)$, she is still faced with the of breaking the ElGamal encryption scheme, i.e. $(y'_3 \cdot J_i^{-1}) \cdot (y'_s)^{-1} \mod p = W_i$. It already knows the ElGamal encryption scheme is IND-CPA secure (Tsionis and Yung, 1998) under the DDH assumption, in which the adversary cannot obtain any bit about the plaintext $W_i = X_i || H(X_i)$.

Furthermore, to compute the hash value $J_i = J(g^{i \cdot r_2} \mod p)$ with the knowledge of the public key $Y = g^r \mod p$ and the value $y'_2 = g^{r_2} \mod p$ is equivalent to solve the CDH assumption, which is weaker than the DDH assumption in the same group (Shoup, 1997). If the DDH assumption is held in the group, then the CDH assumption must be held in that group. Therefore, the security of the proposed scheme can be solely based on the DDH assumption.

To reveal other plaintext $x_j$’s, the adversary cannot compute $J_j$ ($\forall j \neq i$) under the assumption of hash function $J(\cdot)$, since the values of $J_i$ and $J_j$ are nonlinearly related. To meet IND-CCA2, the plaintext $x_i$ is protected under the hash function $H(\cdot)$ to ensure the data integrity and has a data integrity validating step in the decryption algorithm. Without this validating step, the adversary could trivially generate ciphertext for which the corresponding plaintext is unknown. To do this, she just outputs the random strings. In the next section, we give the analyses of the reduction for proving its securities.

5.2. Security Analysis

This section shows that the proposed ElGamal-like PKE is secure in the
IND-CCA2 sense via Proposition 1. Theorem 4 and Theorem 5 shows that there is a plaintext extractor in the ElGamal-extended encryption and is secure in the IND-CPA sense, respectively. Here, we only consider that the plaintext $x$ is smaller than $p$. The sequence number $i$ of $x_i$ presented in the ElGamal-like encryption scheme is omitted.

Proposition 1. $\text{PA} \rightarrow \text{IND-CCA2 (NM-CCA2)}$ in the random oracle model.

A PKE scheme is $\text{PA}$ (Plaint-awareness) is for any ciphertext the adversary produces, s/he must know the corresponding the plaintext. Bellare et al. (Bellare et al., 1998) proved that if a PKE scheme is secure in the $\text{PA}$ sense, then it is secure in the IND-CCA2 (or NM-CCA2) sense in the random oracle model.

Theorem 4 Plaintext extractor $\mathcal{P}$ of $\Pi^\dagger = (\mathcal{K}^\dagger, \mathcal{E}^\dagger, \mathcal{D}^\dagger)$. If there exists a $(t, q_H, q_J)$-adversary $\mathcal{B}$, then there exists a constant $c$ and a $(t', \lambda)$-plaintext extractor $\mathcal{P}$ such that

$$t' = t + q_J q_H (t_E + c) \text{ and } \lambda = 1 - \frac{q_J}{2k} + |H| \cdot \frac{1}{2^k}.$$  

$t_E$ denotes the computational running time of the encryption algorithm $\mathcal{E}$ such that $(y'_1, y'_3 \cdot (Y^{[h_v]2k_1} k_1)^{-1} \mod p) = \mathcal{E}_{pk}(hv, [h_v]2k_1 k_1)$ in the specification of $\mathcal{P}$. $|H|$ denotes the number of pairs $(h, H_v)$ in the set $H_s$.

Proof 4. We construct a plaintext extractor $\mathcal{P}$ as follows:

Extractor: $\mathcal{P}(hH, jJ, C, (y'_1, y'_2, y'_3), pk)$

For $u = 1, \ldots, q_H$ do

For $v = 1, \ldots, q_J$ do

$(y_1, y_3) \leftarrow (y'_1, y'_3 \cdot J^{-1} u \mod p)$

If $(y_1, y_3) = \mathcal{E}_{pk}(h_v, [h_v]2k_1 k_1)$

If $J_u = Y^{[h_v]2k_1 k_1} \mod p$

then $x \leftarrow [h_v]^{k_0}$ and break

Else $x \leftarrow \text{null}$

Return $x$

End.

Let $c$ be the computation time of comparing two strings is equal or not, and some overhead. From the specification of $\mathcal{P}$, it runs within $t + q_J q_H (t_E + c)$.

Since there exists an additional random oracle $J(\cdot)$, $J = \{(j_1, J_1), \ldots, (j_q, J_q)\}$ denotes the set of all $\mathcal{B}$’s queries and the corresponding answers of $J(\cdot)$. Intuitively, the plaintext $x$ together with the random numbers $r_1, r_2$ are inputs to the random oracle $H(\cdot)$. Moreover, all the answers to queries should be obtained by the random oracles in the random oracle model. Furthermore, those
queries and the corresponding answers are recorded in the lists $hH$ and $jJ$. Any generation of valid ciphertext should be obtained via that step. Hence, upon input of the valid ciphertext, $\mathcal{P}$ can find out the corresponding plaintext by watching the lists $hH$ and $jJ$.

Now the probability that $\mathcal{P}$ correctly outputs the plaintext $x$, that is $x = \mathcal{D}^{\mathcal{P}}_{sk}(y_1', y_2', y_3')$. Consider the following events:

Con1 $\land$ Con2: the product of events Con1 and Con2, which is assigned to be true if there exists $(j, J)$ in the list $jJ$ and $(h, H)$ in the list $hH$ such that the conditions $(y_1, y_3) = E_{pk}(h_v||H_v, [h_{v, 2k_1}]^k_v)$ and $j_u = Y([h_{v, 2k_1}]^k_v)$ in the specification of $\mathcal{P}$ hold. Two conditions are separately denoted as Con1 and Con2.

Fail: an event assigned to be true if $x \neq \mathcal{D}^{\mathcal{P}}_{sk}(y_1', y_2', y_3')$.

We now bound the failure probability as follows:

$$
\Pr[\text{Fail}] = \Pr[\text{Fail} | \text{Con1 $\land$ Con2}] \cdot \Pr[\text{Con1 $\land$ Con2}] + \\
\Pr[\text{Fail} | \text{Con1 $\land$ $\neg$Con2}] \cdot \Pr[\text{Con1 $\land$ $\neg$Con2}] + \\
\Pr[\text{Fail} | \text{$\neg$Con1}] \cdot \Pr[\text{$\neg$Con1}] \\
\leq \Pr[\text{Fail} | \text{Con1 $\land$ Con2}] + \Pr[\text{Con1 $\land$ $\neg$Con2}] + \Pr[\text{$\neg$Con1}]
$$

In the following, we upper bound $\Pr[\text{Fail} | \text{Con1 $\land$ Con2}]$, $\Pr[\text{Con1 $\land$ $\neg$Con2}]$, and $\Pr[\text{$\neg$Con1}]$, respectively.

The specification of $\mathcal{P}$ is as follows. If Con1 $\land$ Con2 is true then $\mathcal{P}$ never fails to guess the plaintext $x$ and hence $\Pr[\text{Fail} | \text{Con1 $\land$ Con2}] = 0$.

We further upper bound $\Pr[\text{Con1 $\land$ $\neg$Con2}]$ as follows:

$$
\Pr[\text{Con1 $\land$ $\neg$Con2}] \leq \Pr[\text{Con1 $\land$ $\neg$Con2}]
$$

When $\neg$Con2 is true, there is a $j_u$ in the list $jJ$ such that $(y_1', y_3', J_u^{-1} \mod p) = E_{pk}(h_v||H_v, [h_{v, 2k_1}]^k_v)$. Under the random oracle model assumption, the probability of such $j_u$ is $\frac{1}{2^k}$. The conditional probability $\Pr[\text{Con1 $\land$ $\neg$Con2}]$ is $\frac{1}{2^k}$.

For $\Pr[\text{Fail} | \text{$\neg$Con1}]$, $\neg$Con1 is true and $\mathcal{P}$ outputs null. That is, it guesses $(y_1', y_2', y_3')$ is a invalid ciphertext. Therefore, Fail is true implies $\mathcal{B}$ outputs the valid ciphertext $(y_1', y_2', y_3')$. For a fixed $(y_1', y_2', y_3')$ and $J = J(Y[|h_{v, 2k_1}]^k_v, \mod p)$, let $H_s$ be the set of $(h, H_v)$ such that $(y_1', y_3', J^{-1} \mod p) = E_{pk}(h||H_v, [[h]_{2k_1}]^k_v)$. Then since $(y_1', y_2', y_3') \notin C = \{(y_1', y_2', y_3') | \neg C \}$
and hence $D_{sk}((y'_1, y'_2, J^{-1} \mod p)_i) \neq h||H(h)$ for every $(y'_1, y'_2, y'_3)_i \in C$.

For a fixed $(y'_1, y'_2, y'_3)$ and a fixed $h$, since $B$ doesn’t ask query $h$ to oracle $H(\cdot)$,

$$\Pr[\text{Fail} | \neg \text{Con1}] = \Pr_{H \leftarrow \Omega}[H(h) \in H] = \frac{|H|}{2^r},$$

Obviously, $|H|$ is small. We conclude that $Pr[\text{Fail}] \leq \frac{q_J}{2^k} + \frac{|H|}{2^r}$. Hence, $\epsilon = 1 - Pr[\text{Fail}] = 1 - \left(\frac{q_J}{2^k} + \frac{|H|}{2^r}\right)$.

**Theorem 5 PKE: IND-CPA.** If there exists a $(t, q_H, q_J, \epsilon)$-breaker $A = (A_1, A_2)$ for $\Pi^* = (K^*, E^*, D^*)$ in the IND-CPA sense in the random oracle model, then there exists a constants $c$ and a $(t', \epsilon')$-breaker $A' = (A'_1, A'_2)$ for $\Pi = (K, E, D)$ in the IND-CPA sense in the standard model, where

$$t' = t + q_H \cdot c + q_J \cdot c \text{ and } \epsilon' = \epsilon - \frac{q_H}{2^{(2k_1-2)}}.$$

**Proof 5.** We construct a breaking algorithm $A' = (A'_1, A'_2)$ in the IND-CPA and standard model setting by using $A = (A_1, A_2)$ as an oracle.

Firstly, $A'$ initiates two lists $hH$ and $jJ$, to empty. Basically, when $A$ asks query $h$ and $j$, $A'$ simulates two random oracles $H(\cdot)$ and $J(\cdot)$ as follows: If $h$ has not been asked in the list $hH$, $A'$ provides a random string $H$ of length $\iota$-bit, and adds an entry $(h, H)$ to the list $hH$. Similarly, if $j$ has not been asked in the list $jJ$, $A'$ provides a random string $J$ of length $k$-bit, and adds an entry $(j, J)$ to the list $jJ$. When $A_1$ halts and outputs $(x_0, x_1, \omega)$, $A'_1$ outputs $(x_0||\gamma_0||\beta_0, x_1||\gamma_1||\beta_1, \omega)$ where $\gamma_0, \gamma_1$ are $(2k_1)$-bit random strings and $\beta_0, \beta_1$ are $\iota$-bit random strings.
Adversary: \( A'_1(pk) \)
\( hH, jJ \leftarrow \text{empty} \)
Run \( A_1(pk) \)

Do while \( A_1 \) does not make \( H \)-query \( h \) and \( J \)-query \( j \)

If \( A_1 \) makes \( J \)-query \( j \)

If \( j \not\in jJ \)
\( J \leftarrow R \{0, 1\}^k \)
Put \( (j, J) \) on \( jJ \)
Answer \( J \) to \( A_1 \)
Else \( j \in jJ \)
Answer \( J \) to \( A_1 \)

Else if \( A_1 \) makes \( H \)-query \( h \)

If \( h \not\in hH \)
\( H \leftarrow R \{0, 1\}^i \)
Put \( (h, H) \) on \( hH \)
Answer \( H \) to \( A_1 \)
Else \( h \in hH \)
Answer \( H \) to \( A_1 \) such that \( (h, H) \in hH \)

\( A_1 \) outputs \((x_0, x_1, \omega)\)
\( \gamma_0, \gamma_1 \leftarrow R \{0, 1\}^{2k_1} \)
\( \beta_0, \beta_1 \leftarrow R \{0, 1\}^i \)
Return \((x_0||\gamma_0||\beta_0, x_1||\gamma_1||\beta_1, \omega)\)

End.

Then, outside of \( A' \), the ciphertext \((y_1, y_3) = E_{pk}(x_b||\gamma_b||\beta_b, R)\) is computed by the encryption oracle \( O_{en} \), where \( b \in_R \{0, 1\} \) and \( R \in_R \mathbb{Z}_q \). Finally, \((x_0, x_1, \omega, (y_1, y_3))\) is input to \( A_2 \).

Encryption oracle: \( O_{en}(x_0||\gamma_0||\beta_0, x_1||\gamma_1||\beta_1, pk) \)
\( R \leftarrow R \mathbb{Z}_q \)
\( b \leftarrow R \{0, 1\} \)
\((y_1, y_3) \leftarrow E_{pk}(x_b||\gamma_b||\beta_b, R) \)
Return \((y_1, y_3)\)

End.

\( A'_2 \) chooses a random string \( r_2 \in \mathbb{Z}_q \) and \( k \)-bit random string \( J^* \). Then it sets \( y'_1 = y_1, y'_2 = g^{r_2} \mod p, \) and \( y'_3 = y_3 \cdot J^* \mod p \). Note that \((y'_1, y'_2, y'_3)\) is treated as the ciphertext of \( x_b \).
Cryptanalysis on ElGamal-Like Public-Key Encryption Scheme

Adversary: \( A'_2(x_0||\gamma_0||\beta_0, x_1||\gamma_1||\beta_1, (y_1, y_3)) \)
\( r_2 \leftarrow \mathbb{Z}_q; J^* \leftarrow \{0, 1\}^k \)
\( y'_1 \leftarrow y_1; y'_2 \leftarrow g^{r_2} \mod p; y'_3 \leftarrow y_3 \cdot J^* \mod p \)

Run \( A_2(x_0, x_1, \omega, (y'_1, y'_2, y'_3)) \)

Do while \( A_2 \) does not make \( H \)-query \( h \) and \( J \)-query \( j \)

Ask \( j \leftarrow \text{false} \)
  If \( A_2 \) makes \( J \)-query \( j \)
    If \( j = Y^{r_2} \mod p \)
      Answer \( J^* \) to \( A_2 \)
      Put \((j, J^*)\) on \( jJ \)
      Ask \( j \leftarrow \text{true} \)
    Else if \( j \notin jJ \)
      \( J \leftarrow \mathbb{R} \{0, 1\}^k \)
      Answer \( J \) to \( A_2 \)
    Else \( j \in jJ \)
      Answer \( J \) to \( A_2 \) such that \((j, J) \in jJ \)

  Else if \( A_2 \) makes \( H \)-query \( h \)
    If \( \text{Ask} j = \text{true} \) and \( h = x_0||\gamma_0 \)
      Stop \( A_2 \) and output \( b \)
    Else if \( h \notin hH \)
      \( H \leftarrow \mathbb{R} \{0, 1\}^t \)
      Put \((h, H)\) on \( hH \)
      Answer \( H \) to \( A_2 \)
    Else \( h \in hH \)
      Answer \( H \) to \( A_2 \) such that \((h, H) \in hH \)

\( A_2 \) outputs \( b \)

End.

The argument behind the proof is as follows: When \( A_2 \) asks the query \( j = Y^{r_2} \mod p \), \( A'_2 \) answers \( J^* \) and \( \text{Ask} j \) is set to true. Since the random string \( r_2 \) is chosen by \( A'_2 \), it has the ability to check whether the query \( j \) is equal to \( Y^{r_2} \mod p \) or not. Once \( \text{Ask} j \) is true and \( A_2 \) asks a query \( h = x_0||\gamma_0 \), it is almost equivalent to \( D_{sk}(y_1, y_3) = D_{sk}(y'_1, y'_3 \cdot (J^*)^{-1} \mod p) \), since \( A_2 \) has no clue to \( \gamma_0 \) where \( \bar{b} \) is the complement of bit \( b \). The probability to ask \( h = x_0||\gamma_0 \) is \( \frac{1}{2^t} \) which is negligible. Under the condition \( \text{Ask} j \) is true, \( A'_2 \) can expect that it will output a correct bit \( b \) if \( A_2 \) asks either \( h = x_0||\gamma_0 \) or \( h = x_1||\gamma_1 \). If \( A_2 \) asks neither of them, \( A'_2 \) can expect that \( A_2 \) cannot distinguish \((y'_1, y'_2, y'_3)\) from a correct ciphertext.
To analyze the success probability of $A' = (A'_1, A'_2)$, the definitions of success probabilities of $A' = (A'_1, A'_2)$ and $A = (A_1, A_2)$ in Definition 6 are recalled. Consider the follows events to capture the success probabilities of $A = (A_1, A_2)$ and $A' = (A'_1, A'_2)$

Ask $j$: is true if a $J$-query $j = Y^{x_2} \mod p$ was made by $A_2$.

Ask $b$: is true if a $H$-query $h = x_b \| |_n$ was made by $A_2$.

Ask $b'$: is true if a $H$-query $h = x_b \| |_n$ was made by $A_2$.

The probability of Succ$^{\text{IND-CPA}}_{A', k}$ $(k)$ can be obtained by considering the conditions of the product of events Ask $j \land$ Ask $b$ and its complement. Then,

$$\Pr[\text{Succ}^{\text{IND-CPA}}_{A', k}] = \Pr[\text{Succ}^{\text{IND-CPA}}_{A', k} | \text{Ask} j \land \text{Ask} b] \times \Pr[\text{Ask} j \land \text{Ask} b] + \Pr[\text{Succ}^{\text{IND-CPA}}_{A', k} | \neg \text{Ask} j \land \neg \text{Ask} b] \times \Pr[\neg \text{Ask} j \land \neg \text{Ask} b].$$

The probability of $\neg \text{Ask} j \lor \neg \text{Ask} b$ can be written as,

$$\Pr[\neg \text{Ask} j \lor \neg \text{Ask} b] = \Pr[(-\text{Ask} j \lor \neg \text{Ask} b) \land \text{Ask} b] + \Pr[(-\text{Ask} j \lor \neg \text{Ask} b) \land \neg \text{Ask} b].$$

Then,

$$\Pr[\text{Succ}^{\text{IND-CPA}}_{A', k}] = \Pr[\text{Succ}^{\text{IND-CPA}}_{A', k} | \text{Ask} j \land \text{Ask} b] \times \Pr[\text{Ask} j \land \text{Ask} b] + \Pr[\text{Succ}^{\text{IND-CPA}}_{A', k} | \neg \text{Ask} j \land \neg \text{Ask} b] \times \Pr[\neg \text{Ask} j \land \neg \text{Ask} b] + \Pr[\text{Succ}^{\text{IND-CPA}}_{A', k} | \neg \text{Ask} j \land \text{Ask} b] \times \Pr[\neg \text{Ask} j \land \text{Ask} b] + \Pr[\text{Succ}^{\text{IND-CPA}}_{A', k} | \text{Ask} j \land \neg \text{Ask} b] \times \Pr[\text{Ask} j \land \neg \text{Ask} b].$$

Similarly,

$$\Pr[\text{Succ}^{\text{IND-CPA}}_{A', k}] = \Pr[\text{Succ}^{\text{IND-CPA}}_{A', k} | \neg \text{Ask} j \land \text{Ask} b] \times \Pr[\neg \text{Ask} j \land \text{Ask} b] + \Pr[\text{Succ}^{\text{IND-CPA}}_{A', k} | \neg \text{Ask} j \land \neg \text{Ask} b] \times \Pr[\neg \text{Ask} j \land \neg \text{Ask} b] + \Pr[\text{Succ}^{\text{IND-CPA}}_{A', k} | \text{Ask} j \land \neg \text{Ask} b] \times \Pr[\text{Ask} j \land \neg \text{Ask} b] + \Pr[\text{Succ}^{\text{IND-CPA}}_{A', k} | \text{Ask} j \land \text{Ask} b] \times \Pr[\text{Ask} j \land \text{Ask} b].$$

From the specification of $A'$, we have the following equations,

$$\begin{align*}
\Pr[\text{Succ}^{\text{IND-CPA}}_{A', k} | \text{Ask} j \land \text{Ask} b] &= 1, \\
\Pr[\text{Succ}^{\text{IND-CPA}}_{A', k} | \neg \text{Ask} j \land \neg \text{Ask} b] &= 0, \\
\Pr[\text{Succ}^{\text{IND-CPA}}_{A', k} | \neg \text{Ask} j \land \text{Ask} b] &= 0, \\
\Pr[\text{Succ}^{\text{IND-CPA}}_{A', k} | \text{Ask} j \land \neg \text{Ask} b] &= 0.
\end{align*}$$

Equation (1) and Equation (2) are computed as follows.

$$\Pr[\text{Succ}^{\text{IND-CPA}}_{A', k}] = \Pr[\text{Succ}^{\text{IND-CPA}}_{A', k}].$$
\[\Pr[\text{Succ}^{\text{IND-CPA}}_A(k) | \text{Ask}_j \wedge \text{Ask}_b] \cdot \Pr[\text{Ask}_j \wedge \text{Ask}_b] \geq \Pr[\text{Ask}_j \wedge \text{Ask}_b] = \frac{q_H}{2^{2k_1}}\]
Adversary: \[ A'_1(pk) \]
\[ hH, jJ \leftarrow \text{empty} \]
Run \[ A'_{D, h, J}^D(pk) \]
\begin{itemize}
  \item Do while \( A_1 \) does not make \( H \)-query \( h \), \( J \)-query \( j \), \( D \)-query \((y_1, y_2, y_3)\)
  \begin{enumerate}
    \item If \( A_1 \) makes \( J \)-query \( j \)
      \begin{itemize}
        \item If \( j \not\in jJ \)
          \[ J \leftarrow R \{0, 1\}^k \]
          \[ \text{Put} \ (j, J) \text{ on } jJ \]
          \[ \text{Answer } J \text{ to } A_1 \]
        \item Else \( j \in jJ \)
          \[ \text{Answer } J \text{ to } A_1 \text{ such that } (j, J) \in jJ \]
      \end{itemize}
  \item Else if \( A_1 \) makes \( H \)-query \( h \)
    \begin{itemize}
      \item If \( h \not\in hH \)
        \[ H \leftarrow R \{0, 1\}^l \]
        \[ \text{Put} \ (h, H) \text{ on } hH \]
        \[ \text{Answer } H \text{ to } A_1 \]
      \item Else \( h \in hH \)
        \[ \text{Answer } H \text{ to } A_1 \text{ such that } (h, H) \in hH \]
    \end{itemize}
  \item Else if \( A_1 \) makes \( D \)-query \((y_1, y_2, y_3)\)
    \begin{itemize}
      \item Run \( P(hH, jJ, C, (y_1, y_2, y_3'), pk) \)
      \item \( P \) outputs \( x' \)
      \item Answer \( x' \) to \( A_1 \)
    \end{itemize}
\end{itemize}
\[ A_1 \text{ outputs } (x_0, x_1, \omega) \]
\[ \gamma_0, \gamma_1 \leftarrow R \{0, 1\}^{2k_1} \]
\[ \beta_0, \beta_1 \leftarrow R \{0, 1\}^l \]
\[ \text{Return } (x_0 \| \gamma_0 \| \beta_0, x_1 \| \gamma_1 \| \beta_1, \omega) \]

End.

Encryption oracle: \[ O_e(x_0 \| \gamma_0 \| \beta_0, x_1 \| \gamma_1 \| \beta_1, pk) \]
\[ b \leftarrow R \{0, 1\} \]
\[ (y_1, y_3) \leftarrow E_{pk}(x_b \| \gamma_0 \| \beta_0, R) \]
\[ \text{Return } (y_1, y_3) \]

End.
Adversary: \( A'_2(x_0||y_0||\beta_0, x_1||\gamma_1||\beta_1, \omega, (y_1, y_3)) \)
\( r_2 \leftarrow \mathbb{Z}_q; J^* \leftarrow \{0,1\}^k \)
\( y'_1 \leftarrow y_1; y'_2 \leftarrow g^{r_2} \text{ mod } p; y'_3 \leftarrow y_3 \cdot J^* \text{ mod } p \)
Run \( A'_2^{D_0,H,J}(x_0, x_1, \omega, (y'_1, y'_2, y'_3)) \)
\( C \leftarrow (y'_1, y'_2, y'_3) \)

\[ \text{Do while } A_2 \text{ does not make } H\text{-query } h \text{ and } J\text{-query } j \]
\[ D\text{-query } (y_1, y_2, y_3)' \]
\[ \text{Ask } j \leftarrow \text{false} \]
\[ \text{If } A_2 \text{ makes } J\text{-query } j \]
\[ \text{If } j = Y^{r_2} \text{ mod } p \]
\[ \text{Answer } J^* \text{ to } A_2 \]
\[ \text{Put } (j, J^*) \text{ on } jJ \]
\[ \text{Ask } j \leftarrow \text{true} \]
\[ \text{Else if } j \notin jJ \]
\[ J \leftarrow \{0,1\}^k \]
\[ \text{Answer } J \text{ to } A_2 \]
\[ \text{Else } j \in jJ \]
\[ \text{Answer } J \text{ to } A_2 \text{ such that } (j, J) \in jJ \]
\[ \text{Else if } A_2 \text{ makes } H\text{-query } h \]
\[ \text{If } \text{Ask } j = \text{true and } h = x_0||\gamma_h \]
\[ \text{Stop } A_2 \text{ and output } b \]
\[ \text{Else if } h \notin hH \]
\[ H \leftarrow \{0,1\}^\iota \]
\[ \text{Put } (h, H) \text{ on } hH \]
\[ \text{Answer } H \text{ to } A_2 \]
\[ \text{Else } h \in hH \]
\[ \text{Answer } H \text{ to } A_2 \text{ such that } (h, H) \in hH \]
\[ \text{Else if } A_2 \text{ makes } D\text{-query } (y_1, y_2, y_3)' \]
\[ \text{Run } P(hH, jJ, C, (y_1, y_2, y_3)', pk) \]
\[ P \text{ outputs } x' \]
\[ \text{Answer } x' \text{ to } A_1 \]
\[ A_2 \text{ outputs } b \]

Return \( b \)

End.

5.3. Performance Analysis

In this section, the computational complexity of the ElGamal encryption scheme with that of the ElGamal-like encryption scheme is compared. Since the time for computing a modular exponentiation computation is much larger than other
operations (modular multiplication computation, modular addition computation, hash function), the following descriptions only compare the number of modular exponentiation computations.

Assume that the whole plaintext $x$ with length $n \cdot k$ is divided into $x_1, x_2, \cdots, x_n$, and the length of each $x_i$ is $k$. To encrypt $x$, the ElGamal encryption scheme requires requires $2n$ modular exponentiation computations. The computational complexity of decrypting requires $n$ modular exponentiation computations.

For the same plaintext $x$ with the length $n \cdot k$ in our ElGamal-like encryption scheme, the maximal length of plaintext is limited by $k_0$. The number of divisions is $n \cdot k \over k_0 = n + \left\lceil n \cdot (2k_1 + i) \right\rceil$. Let $n' = n + \left\lceil n \cdot (2k_1 + i) \right\rceil$. To encrypt $x_1$, the ElGamal-extended scheme requires 4 modular exponentiation computations. To derive the plaintext $x_1$, it requires 2 modular exponentiation computations. To encrypt other $n' - 1$ plaintexts $x_2, \cdots, x_{n'}$, it is not necessary to compute the values $y_1 = g^{r_1} \mod p$, $y_2 = g^{r_2} \mod p$, $Y^{r_1} \mod p$, and $Y^{r_2} \mod p$ again. Hence, 4 modular exponentiation computations is only needed for $x_1$. The total computational complexity of encrypting $x$ requires 4 modular exponentiation computations. To decrypt other $n' - 1$ ciphertexts $(y_3, 2, \cdots, y_{n'}, n')$, the values $y_1^{r_1} \mod p$ and $y_2^{r_1} \mod p$ have also been computed. The total computational complexity of decrypting requires 2 modular exponentiation computations.

6. Discussion and Conclusion
The ElGamal PKE has been proven to be secure in the IND-CPA sense in the standard model if the operation is in $QR_p$ (Tsiounis and Yung, 1998). The IND-CPA sense is considered as a basic requirement for most provably secure PKEs. In many applications, plaintexts may contain information which can be guessed easily such as in a BUY/SELL instruction to a stock broker. In this paper, we precisely show that the ElGamal is insecure in the IND-CPA sense if the operation is not $QR_p$. For Wang et al.’s improved ElGamal-like PKE, we give two simple examples to prove it is insecure in the IND-CPA sense either operated in $QR_p$ or not (employ the key generation $K'$ or $\hat{K}$). Besides, the cryptosystem has the probability to be crashed when $Y^{r_1} \oplus (Y^{r_2})' \mod p = 0$. Since the exclusive-or operation is not suitable for the group operation, the computed values cannot be expected in that group.

The motivation for encrypting large messages in PKEs is practical, since they have bad performance as compared to symmetric encryption schemes. The proposed ElGamal-like encryption scheme for encrypting large messages is easily proven IND-CCA2 security in the random oracle model. Obviously, if the hash functions $H(a)$ and $J(a)$ are implemented by $g^a \mod p$ (universal one-way
hash functions) rather than MD5 or SHA (collision-resistant hash functions), the ElGamal-extended encryption scheme can be proven in the standard model. However, it is contradiction for encrypting large messages efficiently.

Acknowledgements
The authors wish to thank many anonymous referees for their suggestions to improve this paper. Part of this research was supported by the National Science Council, Taiwan, R.O.C., under contract no. NSC100-2221-E-018-025 and NSC100-2622-E-018-004-CC3.

REFERENCES
Ting Yi Chang et al.


**Ting Yi Chang** received his M.S. from the Graduate Institute of Computer Science and Information Engineering from the Chaoyang University of Technology in 2003, and a Ph.D in the Department of Computer Science from the National Chiao Tung University, Taiwan in 2006. Currently, he is an Associate Professor with the Graduate Institute of e-Learning, National Changhua University, Taiwan. His current research interests include artificial intelligence, e-Learning, information security, cryptography, and mobile communications.
Min-Shiang Hwang received the B.S. in Electronic Engineering from the National Taipei Institute of Technology, Taipei, Taiwan, Republic of China (ROC), in 1980; the M.S. in Industrial Engineering from the National Tsing Hua University, Taiwan, in 1988; and a Ph.D. in Computer and Information Science from the National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at the National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in field "Electronic Engineer" in 1988. He also passed the National Telecommunication Special Examination in field "Information Engineering", qualified as advanced technician the first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, ROC. He was also the chairman of the Department of Information Management, Chaoyang University of Technology (CYUT), Taiwan, during 1999-2002. He was a professor and chairman of the Graduate Institute of Networking and Communications, CYUT, during 2002-2003. He was a professor and chairman of the Department of Management Information Systems, National Chung Hsing University (NCHU), during 2005-2009. He was an outstanding professor of the department of Management Information Systems, NCHU, during 2007-2011. He obtained the 1997, 1998, 1999, 2000, and 2001 Outstanding Research Award of National Science Council of the Republic of China. He is currently a Chair Professor of the department of Computer Science & Information Engineering, Asia University. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include Information Security, electronic commerce, database and data security, cryptography, image compression, and mobile computing. Dr. Hwang has published over 170+ articles on the above research fields in international journals.

Wei-Pang Yang was born on May 17, 1950 in Hualien, TAIWAN. He received the B.S. degree in mathematics from National Taiwan Normal University in 1974, and the M.S. and Ph.D. degrees from the National Chiao Tung University in 1979 and 1984, respectively, both in computer engineering. Since August 1979, he has been on the faculty of the Department of Computer Science and Information Engineering at National Chiao Tung University, Hsinchu, Taiwan. In the academic year 1985-1986, he was awarded the National Postdoctoral Research Fellowship and was a visiting scholar at Harvard University. From 1986 to 1987. He was the Director of the Computer Center of National Chiao Tung University. In August 1988, he joined the Department of Computer and Information Science at National Chiao Tung University, and acted as the Head of the Department for one year. Then he went to IBM Almaden Research Center in San Jose, Cali-
fornia for another one year as visiting scientist. From 1990 to 1992, he was the Head of the Department of Computer and Information Science again. He was the visiting scholar of the University of Washington in Seattle for one year in 1996-97. He was the Director of University Library from 1988-2004. Back to home town Hualien. Since 2004 he has transferred to National Dong Hwa University at Hualien (the most beautiful county in Taiwan and Wei-Pang was born here). He acted as the Head of the Department of Information Management from 2004 to 2005, the Dean of College of Management from 2005 to 2007, and Now the Dean of Academic Affairs of NDHU. Now he is the Vice President of NDHU. His research interests include database theory and application, information retrieval, data mining, digital library, and digital museum. Dr. Yang is a senior member of IEEE, and a member of ACM. He was the winner of the 1988, and 1992 AceR Long Term Award for Outstanding M.S. Thesis Supervision, 1993 AceR Long Term Award for Outstanding Ph. D. Dissertation Supervision, and the winner of 1990 Outstanding Paper Award of the Computer Society of the Republic of China. He also obtained the Outstanding Research Award of National Science Council of Taiwan. And he was the winner of the Outstanding Project Award, National Chiao Tung University, 2004. And he also obtained the Outstanding Research Award of National Dong Hwa University, 2010-2012.