INFORMATICA, 2012, Vol. 23, No. 4, 537–562

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

Ting-Yi CHANG[1], Min-Shiang HWANG[2] *, Wei-Pang YANG[3]

[1]*Department of Industrial Education and Technology, National Changhua University of Education No. 1, Jin-De Road, Changhua City, Taiwan, R.O.C.*

[2]*Department of Computer Science & Information Engineering, Asia University 1500, Lioufeng Rd., Wufeng, Taichung, Taiwan, R.O.C.*

[3]*Department of Information Management, National Dong Hwa University 1, Sec. 2, Da Hsueh Rd., Shou-Feng, Hualien, Taiwan, R.O.C.*

e-mail: tychang@cc.ncue.edu.tw, mshwang@mail.cyut.edu.tw, wpyang@mail.ndhu.edu.tw

Received: August 2010; accepted: September 2012

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.

Keywords: public-key encryption, cryptanalysis, chosen-plaintext attack, adaptive chosen-chiphertext attack, chosen-ciphertext attack, Diffie–Hellman problem, indistinguishable.

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, 2009, 2010; Chmielowiec, 2010; Hwang *et al.*, 2003; Wang and Hy, 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,

*Corresponding author.

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.* (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.

$$\begin{array}{ccccc}
\mathsf{NM\text{-}CPA} & \longleftarrow & \mathsf{NM\text{-}CCA1} & \overset{\longleftarrow}{\not\longrightarrow} & \mathsf{NM\text{-}CCA2}\\
\downarrow & \not\searrow\ \not\nwarrow & \downarrow & & \downarrow\ \uparrow \\
\mathsf{IND\text{-}CPA} & \longleftarrow & \mathsf{IND\text{-}CCA1} & \longleftarrow & \mathsf{IND\text{-}CCA2}
\end{array}$$

For $\mathbb{A}, \mathbb{B}$ $\in$GOAL-ATK, "$\mathbb{A} \rightarrow \mathbb{B}$" denotes $\mathbb{A}$ implies $\mathbb{B}$, which means if a PKE is secure in the sense of $\mathbb{A}$, it is also secure in the sense of $\mathbb{B}$. "$\mathbb{A} \not\rightarrow \mathbb{B}$" denotes $\mathbb{A}$ doesn't imply $\mathbb{B}$, which means if a PKE is secure in the sense of $\mathbb{A}$, it is not always secure in the sense of $\mathbb{B}$.

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 (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 (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.* (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 shows the different assumptions and GOAL-ATK among some related schemes. As we realize it is not pratical 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.* (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

Table 1

Assumptions and security notations of some related schemes

Schemes	Assumptions	GOAL-ATK
ElGamal in QR_p (Tsiounis and Yung, 1998)	DDH problem	IND-CPA
Tsiouns-Yung (Tsiounis and Yung, 1998)	DDH problem, RO	IND-CCA2
Shoup–Gennaro (Shoup and Gennaro, 1998)	DDH problem, RO	IND-CCA2
Cramer–Shoup (Cramer and Shoup, 1998)	DDH problem, UOWHF	IND-CCA2
Pointcheval (Pointcheval, 1999)	DRSA problem, RO	IND-CCA2
Paillier–Pointcheval (Paillier and Pointcheval, 1999)	DCR problem, DPDL problem, RO	IND-CCA2
Hwang *et al.* (Hwang *et al.*, 2002)	DDH problem	IND-CPA
Bellare–Rogaway (Bellare and Rogaway, 1994)	Deterministic trap-door partial-domain one-way permutations, RO	IND-CCA2
Fujisaki–Okamoto (Fujisaki *et al.*, 2001)	Probabilistic trap-door one-way functions, RO	IND-CCA2

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

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. 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 cann'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 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 ElGmal-like PKE is insecure in the IND-CPA sense in QR_p and not in 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 $k \geqslant k_0,\ \varepsilon(k) < 1/P(k)$.

DEFINITION 2. Let $\mathcal{A}$ be a probabilistic algorithm and let $\mathcal{A}(x_1, x_2, \ldots; r)$ be the result of running $\mathcal{A}$ on input $x_1, x_2, \ldots$ and coins r. We let $y \leftarrow \mathcal{A}(x_1, x_2, \ldots)$ denote the experiment of choosing r at random and letting y be $\mathcal{A}(x_1, x_2, \ldots; r)$. If S is a finite set, let $x \leftarrow_R S$ 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 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 either the system is 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 $x = y^2 \bmod n$. 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 $\mathrm{QNR}_n = \mathbb{Z}_n^*/\mathrm{QR}_n$.

DEFINITION 5. Let p be a prime > 2, and let $x \in \mathbb{Z}$ be prime to p.

$$\left(\frac{x}{p}\right) := \begin{cases} +1, & if\ [x] \in \mathrm{QR}_p, \\ -1, & if\ [x] \in \mathrm{QNR}_p, \end{cases}$$

is called the Legendre symbol of $x \bmod p$.

DEFINITION 6. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a pair of probabilistic algorithms, say Adversary for $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. For ATK={CPA, CCA1, CCA2} and $k \in \mathbb{N}$, denote the success event of $\mathcal{A}$ for Π by

$$\mathsf{Succ}^{\mathsf{ATK}}_{\mathcal{A},\Pi}(k) = \big[(pk, sk) \leftarrow \mathcal{K}(1^k); (x_0, x_1, state) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(pk); b \leftarrow_R \{0,1\}; \\ y \leftarrow \mathcal{E}_{pk}(x_b) : \mathcal{A}_2^{\mathcal{O}_2}(x_0, x_1, state, y) = b\big],$$

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, $\mathcal{O}_1(\cdot),\ \mathcal{O}_2(\cdot)$ are defined as follows:

–If ATK=CPA then $\mathcal{O}_1(\cdot) = \mathrm{null}$ and $\mathcal{O}_2(\cdot) = \mathrm{null}$;
–If ATK=CCA1 then $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ and $\mathcal{O}_2(\cdot) = \mathrm{null}$;
–If ATK=CCA2 then $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ and $\mathcal{O}_2(\cdot) = \mathcal{D}_{sk}(\cdot)$.

We denote the advantage of $\mathcal{A}$ for Π as

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

We say that Π is secure in the IND-ATK sense if for any adversary $\mathcal{A}$ being polynomial-time in k, $\mathsf{Adv}^{\mathsf{ATK}}_{\mathcal{A},\Pi}(k)$ 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 $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the ElGamal PKE.

– Key generation algorithm $\mathcal{K}$: $(pk, sk) \leftarrow \mathcal{K}(1^k)$, $pk = (p, g, Y)$ and $sk = s$, where $Y = g^s \bmod p$, $|p| = k$, $s \in \mathbb{Z}_p^*$, and $\#\langle g \rangle = p$. Let $\mathbb{G}_p$ be a group of prime order p of the multiplicative group $\mathbb{Z}_p^*$.
– Encryption algorithm $\mathcal{E}$:

$$(y_1, y_2) = \mathcal{E}_{pk}(x; r) = \big(g^r \bmod p, x \cdot Y^r \bmod p\big),$$

where message $x \in \{0,1\}^k$ and $r \leftarrow_R \{0,1\}^k$.
– Decryption algorithm $\mathcal{D}$:

$$x = \mathcal{D}_{sk}(y_1, y_2) = y_2 \cdot \big(y_1^s\big)^{-1} \bmod p.$$

3.2. *Security Analysis*

We can see that g is a primitive root of $\mathbb{G}_p$ by employing the key generation algorithm $\mathcal{K}$ 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 $x \in \mathrm{QR}_p$ if and only if $x = g^a \bmod p$ some even number a, $0 \leqslant a < p-1$.*

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 $[xy] \in \mathrm{QR}_p$ if and only if either both $[x],\ [y] \in \mathrm{QR}_p$ or both $[x],\ [y] \in \mathrm{QNR}_p$.

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

$$\epsilon = 1 \quad \textit{and} \quad t \leqslant t_1 + 3 \cdot t_{\mathrm{QR}}.$$

Proof. We construct a breaking algorithm $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ for $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ as follows.

Adversary: $\mathcal{A}_1(pk)$
Obtain $\{x_0, x_1\}$, where $x_0 \in \mathrm{QR}_p$ and $x_1 \in \mathrm{QNR}_p$
Return $(x_0, x_1, state)$
End.
Encryption oracle: $\mathcal{O}_{en}(x_0, x_1, pk)$
$b \leftarrow_R \{0,1\}$
$(y_1, y_2) = \mathcal{E}_{pk}(x_b; r) = (g^r \bmod p, x_b \cdot Y^r \bmod p)$
End.

Adversary: $\mathcal{A}_2(x_0, x_1, state, (y_1, y_2))$

Case 1: $Y \in \mathrm{QR}_p$ and $y_1 \in \{\mathrm{QR}_p, \mathrm{QNR}_p\}$

If $y_2 \in \mathrm{QR}_p$, then outputs 0

If $y_2 \in \mathrm{QNR}_p$, then outputs 1

Case 2: $y_1 \in \mathrm{QR}_p$ and $Y \in \mathrm{QNR}_p$

If $y_2 \in \mathrm{QR}_p$, then outputs 0

If $y_2 \in \mathrm{QNR}_p$, then outputs 1

Case 3: $Y \in \mathrm{QNR}_p$, $y_1 \in \mathrm{QNR}_p$

If $y_2 \in \mathrm{QNR}_p$, then outputs 0

If $y_2 \in \mathrm{QR}_p$, then outputs 1

End.

We now analyze the successful probability of adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. We define the following events. E_1 be the event $(Y \in \mathrm{QR}_p) \wedge (y_1 \in \{\mathrm{QR}_p, \mathrm{QNR}_p\})$, E_2 be the event $(y_1 \in \mathrm{QR}_p) \wedge (Y \in \mathrm{QNR}_p)$ and E_3 be the event $(Y \in \mathrm{QNR}_p) \wedge (y_1 \in \mathrm{QNR}_p)$.

Let b' be the output of $\mathcal{A}_2$. For Case 1, $Y = g^s \in \mathrm{QR}_p$. By Lemma 1, s is even, no matter what $y_1 \in \mathrm{QR}_p$, or $y_1 \in \mathrm{QNR}_p$, we know that $Y^r = g^{sr} \in \mathrm{QR}_p$. We see that $\mathcal{A}_2$ will output the correct b'=0 (b'=1) if and only if $y_2 \in \mathrm{QR}_p$ ($y_2 \in$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 Pr[$b = b'|\mathsf{E}_1$]=1 and the probability Pr[E_1] $= 1/2$. For the same reason, in Case 2, the condition probability Pr[$b = b'|\mathsf{E}_2$]=1. Note that $(y_1 \in \mathrm{QR}_p) \wedge (Y \in \mathrm{QR}_p)$ is included in the event E_1 and the probability Pr[E_1] $= 1/4$. For Case 3, $Y \in \mathrm{QNR}_p$ and $y_1 \in \mathrm{QNR}_p$, by Lemma 1, s and r are odd, $Y^r = g^{sr} \in \mathrm{QNR}_p$. $\mathcal{A}_2$ will output the correct b'=0 (b'=1) if and only if $y_2 \in \mathrm{QR}_p$ ($y_2 \in \mathrm{QNR}_p$). Thus, the condition probability Pr[$b = b'|\mathsf{E}_3$] $= 1$ and the probability Pr[E_3] $= 1/4$. By the law of total probability,

$$\begin{aligned}
\Pr\big[\mathsf{Succ}^{\mathsf{CPA}}_{\mathcal{A},\Pi}(k)\big] &= \Pr\big[b = b'\big] \\
&= \sum_{i=1}^{3} \Pr\big[b = b'|\mathsf{E}_i\big] \cdot \Pr[\mathsf{E}_i] \\
&= 1 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} + 1 \cdot \frac{1}{4} \\
&= 1,
\end{aligned}$$

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

Thus, we have the ability to distinguish the distinct plaintext 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 $\mathbb{G}_q$ of prime order q of the multiplicative group $\mathbb{Z}_p^*$ and g of $\mathbb{G}_q$ are defined. In other words, the key generation $\mathcal{K}$ should be modified as $\widehat{\mathcal{K}}$.

– Key generation $\widehat{\mathcal{K}}$: $(pk, sk) \leftarrow \widehat{\mathcal{K}}(1^k)$, $pk = (p, g, Y)$ and $sk = (p, g, s)$, where $Y = g^s \bmod p$, $|p| = k$, $p = 2q + 1$, $\#\langle h\rangle = p$, $g = h^2 \bmod p$, $s \in \mathbb{Z}/q\mathbb{Z}$, and $\#\langle g\rangle = q$.

Since g generates all the quadratic residues in QR_p and the message for encrypting is needed to be a QR_p, the algorithm $\mathcal{A}$ does not work.

A value is determined whether it is in 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 QR_p or not. Let t_1 be the time of choosing two messages $x_0 \in \mathrm{QR}_p$ and $x_1 \in \mathrm{QNR}_p$. Then, from the specification of $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, it runs within at most 3 times t_{QR} in Case 2 or Case 3. Hence, $t \leqslant t_1 + 3 \cdot t_{\mathrm{QR}}$ and it is in a polynomial time.

4. Analysis of ElGamal-Like PKE Scheme

The ElGamal should employ the key generation algorithm $\widehat{\mathcal{K}}$ 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 $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ be Wang et al.'s ElGamal-like PKE.

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

$$(y_1, y_2, y_{3,i}) = \mathcal{E}'_{pk}(x_i; r_1; r_2) = \big(g^{r_1} \bmod p, g^{r_2} \bmod p, x_i \\ \times \big(Y^{r_1} \oplus \big(Y^{r_2}\big)^i\big) \bmod p\big),$$

  where message $x \in \{0,1\}^{>k}$, x is divided into $x_1, x_2, \ldots, x_n$ ($|x_1| = |x_2| = \cdots = |x_{n-1}|$, $n = \lceil |x|/k \rceil$, $|x_n| = |x| \bmod k$, 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 $\mathcal{D}'$:

$$x_i = \mathcal{D}'_{sk}(y_1, y_2, y_{3,i}) = y_{3,i} \cdot \big(y_1^s \oplus \big(y_2^s\big)^i\big)^{-1} \bmod p, \\ x = x_1 x_2 \ldots x_n.$$

  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 $\mathcal{E}'$:

$$(y_1, y_2, y_3) = \mathcal{E}'_{pk}(x; r_1; r_2) \\ = \big(g^{r_1} \bmod p, g^{r_2} \bmod p, x \times \big(Y^{r_1} \oplus Y^{r_2}\big) \bmod p\big),$$

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

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

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

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', ϵ')*-breaker for* Π' *in* IND-CPA *if* $\mathsf{Adv}^{\mathsf{CPA}}_{\mathcal{A}',\Pi'}(k) \geqslant \epsilon'$ *with the event* Fail *does not occur, and* $\mathcal{A}'$ *runs within at most running time* t'*, where*

$$\epsilon' = 1 \quad and \quad t' \leqslant t_1 + 3 \cdot t_{\mathrm{QR}}.$$

Proof. We give a simple example and then analyze the results as follows. In the key generation algorithm $\mathcal{K}'$, 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 $2^{T+1} = 2 \bmod p - 1$ is as large as possible. In this example, $2^j \bmod p$ generate $\{5, 7\}$ for all integer j in $[1, 6]$ since the smallest positive integer such that $2^{T+1} = 2 \bmod p - 1$ is 2. Consider the sets $\mathrm{QR}_p = \{1, 2, 4\}$ and $\mathrm{QNR}_p = \{3, 5, 6\}$. By Lemma 1, the following situations $(\bmod\ p$ is abridged) are considered.

Situation 1: $Y^{r_1} \in \mathrm{QR}_p$ and $Y^{r_2} \in \mathrm{QR}_p$

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: $Y^{r_1} \in \mathrm{QR}_p$ and $Y^{r_2} \in \mathrm{QNR}_p$

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: $Y^{r_1} \in \mathrm{QNR}_p$ and $Y^{r_2} \in \mathrm{QR}_p$

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: $Y^{r_1} \in \mathrm{QNR}_p$ and $Y^{r_2} \in \mathrm{QNR}_p$

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 $\mathcal{E}'$, the value of $y_3 = x \cdot (Y^{r_1} \oplus Y^{r_2})$ is equal to 0.

The encrypt algorithm $\mathcal{E}'$ is failed, together with the decrypt algorithm $\mathcal{D}'$. Obviously, the probability of $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ crashed in the above situations is $1/3$, denoted as $\Pr[\mathsf{Fail}] = 1/3$.

If the encryption algorithm $\mathcal{E}'$ 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 $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ in the $\mathsf{IND\text{-}CPA}$ sense for $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$.

Adversary: $\mathcal{A}'_1(pk)$
Obtain $\{x_0, x_1\}$, where $x_0 \in \mathrm{QR}_p$ and $x_1 \in \mathrm{QNR}_p$
Return $(x_0, x_1, state)$
End.

Encryption oracle: $\mathcal{O}_{en}(x_0, x_1, pk)$
$b \leftarrow_R \{0,1\}$
$(y_1, y_2, y_3) = \mathcal{E}'_{pk}(x_b; r_1; r_2) = (g^{r_1}, g^{r_2}, x_b \cdot (Y^{r_1} \oplus Y^{r_2}))$
End.

Adversary: $\mathcal{A}'_2(x_0, x_1, state, (y_1, y_2, y_3))$
Case 1: $Y \in \mathrm{QR}_p$ // $Y^{r_1} \oplus Y^{r_2} \in \mathrm{QNR}_p$
If $y_3 \in \mathrm{QR}_p$, then outputs 1
If $y_3 \in \mathrm{QNR}_p$,then outputs 0
Case 2: $Y \in \mathrm{QNR}_p$ and $y_1 \in \mathrm{QR}_p$ and
$y_2 \in \mathrm{QR}_p$ //$Y^{r_1} \oplus Y^{r_2} \in \mathrm{QNR}_p$
If $y_3 \in \mathrm{QR}_p$, then outputs 1
If $y_3 \in \mathrm{QNR}_p$, then outputs 0
Case 3: $Y \in \mathrm{QNR}_p$ and
$y_1 \in \mathrm{QNR}_p$ and $y_2 \in \mathrm{QR}_p$ //$Y^{r_1} \oplus Y^{r_2} \in \mathrm{QR}_p$
If $y_3 \in \mathrm{QR}_p$, then outputs 0
If $y_3 \in \mathrm{QNR}_p$, then outputs 1
Case 4: $Y \in \mathrm{QNR}_p$ and
$y_1 \in \mathrm{QR}_p$ and $y_2 \in \mathrm{QNR}_p$ //$Y^{r_1} \oplus Y^{r_2} \in \mathrm{QR}_p$
If $y_3 \in \mathrm{QR}_p$, then outputs 0
If $y_3 \in \mathrm{QNR}_p$, then outputs 1
Case 5: $Y \in \mathrm{QNR}_p$ and
$y_1 \in \mathrm{QNR}_p$ and $y_2 \in \mathrm{QNR}_p$ //$Y^{r_1} \oplus Y^{r_2} \in \mathrm{QNR}_p$
If $y_3 \in \mathrm{QR}_p$, then outputs 1
If $y_3 \in \mathrm{QNR}_p$, then outputs 0
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., Fail does not occur. By the multiplicative property of Legendre symbol,

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

the conditional probability $\Pr[\mathsf{Adv}^{\mathsf{CPA}}_{\mathcal{A}',\Pi'}(k) | \neg\mathsf{Fail}]$ is equal to 1 and $\mathsf{Adv}^{\mathsf{CPA}}_{\mathcal{A}',\Pi'}(k) =$

$2 \cdot \Pr[\mathsf{Succ}^{\mathsf{CPA}}_{\mathcal{A}',\Pi'}(k)|\neg\mathsf{Fail}] - 1 = 1$. For the same reason, from the specification of $\mathcal{A}'$, it runs within at most $t' \leqslant t_1 + 3 \cdot t_{\mathrm{QR}}$.

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

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

$$\epsilon'' = 1 \quad and \quad t'' \leqslant t_1 + t_{\mathrm{QR}}.$$

We also give an example for the key generation algorithm $\widehat{\mathcal{K}}$, where $q = 3$, $p = 2q + 1 = 7$, $h = 5$, $g = h^2 \bmod p = 4$. Obviously, $g \in \mathrm{QR}_p$, therefore, the group is in QR_p, where $\mathrm{QR}_p = \{1, 2, 4\}$. The value of $Y^{r_1} \oplus Y^{r_2} \bmod p$ are in the set S_1 as the same as in Situation 1 of Theorem 2. $\Pi'' = (\widehat{\mathcal{K}}, \mathcal{E}', \mathcal{D}')$ has the probability to fail as follows:

$$\begin{aligned}
\Pr[\mathsf{Fail}] &= \Pr\big[\mathsf{Fail}|Y^{r_1} \in \mathrm{QR}_p\big] \cdot \Pr\big[Y^{r_1} \in \mathrm{QR}_p\big] \\
&= \frac{3}{9} \cdot 1 \\
&= \frac{1}{3}.
\end{aligned}$$

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

Proof.

Adversary: $\mathcal{A}''_1(pk)$
Obtain $\{x_0, x_1\}$, where $x_0 \in \mathrm{QR}_p$ and $x_1 \in \mathrm{QNR}_p$
Return $(x_0, x_1, state)$
End.
Encryption oracle: $\mathcal{O}_{en}(x_0, x_1, pk)$
$b \leftarrow_R \{0, 1\}$
$(y_1, y_2, y_3) = \mathcal{E}'_{pk}(x_b; r_1; r_2) = (g^{r_1}, g^{r_2}, x_b \cdot (Y^{r_1} \oplus Y^{r_2}))$
End.
Adversary: $\mathcal{A}''_2(x_0, x_1, state, (y_1, y_2, y_3))$
Case 1: If $y_3 \in \mathrm{QR}_p$, then outputs 1
Case 2: If $y_3 \in \mathrm{QNR}_p$, 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 $x_0 \in \mathrm{QR}_p$ or $x_1 \in \mathrm{QNR}_p$, according to the Legendre symbol $(\frac{y_3}{p})$. This forms Case 1 and Case 2 of $\mathcal{A}''_2$, respectively. The advantage of $\mathcal{A}''$ for Π'' is $\mathsf{Adv}^{\mathsf{CPA}}_{\mathcal{A}'',\Pi''}(k) = 2 \cdot \Pr[\mathsf{Succ}^{\mathsf{CPA}}_{\mathcal{A}'',\Pi''}(k)|\neg\mathsf{Fail}] - 1 = 1$.

From the specification of $\mathcal{A}''$, it runs within at most $t'' \leqslant t_1 + t_{\mathrm{QR}}$. Obviously, the both breaking algorithms $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ and $\mathcal{A}'' = (\mathcal{A}''_1, \mathcal{A}''_2)$ are in a polynomial time in Theorems 4.2 and 4.3, respectively.

We can see that no matter what Wang *et al.*'s ElGamal-like PKE employs $\mathcal{K}'$ or $\widehat{\mathcal{K}}$, 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 Fail will decrease when we chose a large prime q or p (the security parameter k), for both $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ and $\Pi'' = (\widehat{\mathcal{K}}, \mathcal{E}', \mathcal{D}')$, the values after exclusive-or operation may not in the group $\mathbb{G}_p$ and $\mathbb{G}_q$, 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 $\Pi^\dagger = (\mathcal{K}^\dagger, \mathcal{E}^\dagger, \mathcal{D}^\dagger)$ be the ElGamal-extended encryption scheme.

- Key generation algorithm $\mathcal{K}^\dagger$: $(pk, sk) \leftarrow \mathcal{K}^\dagger(1^k)$, $pk = (p, g, Y)$ and $sk = s$, where $Y = g^s \bmod p$, $|p| = k$, $p = 2q + 1$, $\#\langle h\rangle = p$, $g = h^2 \bmod p$, $s \in \mathbb{Z}/q\mathbb{Z}$, and $\#\langle g\rangle = q$. Let $k = k_0 + 2k_1 + \iota$.
- Hash functions H and J: H: $\{0,1\}^{k_0+2k_1} \to \{0,1\}^\iota$, J: $\{0,1\}^k \to \{0,1\}^k$.
- Encryption algorithm $\mathcal{E}^\dagger$:

$$(y'_1, y'_2, y'_{3,i}) = \mathcal{E}^\dagger_{pk}(x_i; r_1; r_2),$$

  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 $J_i = J(Y^{i \cdot r_2} \bmod p)$.

3. Compute $(y_1, y_{3,i}) = (g^{r_1} \bmod p, (X_i||H(X_i)) \cdot Y^{r_1} \bmod p)$.
4. Compute $(y'_1, y'_2, y'_{3,i}) = (y_1, g^{r_2} \bmod p, y_{3,i} \cdot J_i \bmod p)$.

– Decryption algorithm $\mathcal{D}^\dagger$:

$$x_i = \mathcal{D}^\dagger_{sk}(y'_1, y'_2, y'_{3,i}),$$

1. Compute $J_i = J(y_2'^{i \cdot s} \bmod p)$.
2. Compute $W_i = (y'_{3,i} \cdot J_i^{-1}) \cdot (y_1'^{s})^{-1} \bmod p$.
3. Output

$$\begin{cases} [W_i]^{k_0}, & \text{if } H\big([W_i]^{k_0+2k_1}\big) = [W_i]_\iota \\ \text{null}, & \text{otherwise.} \end{cases}$$

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 $x_1, x_2, \dots, x_n$.

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 $J(Y^{i \cdot r_2} \bmod p)$, she is still faced with the of breaking the ElGamal encryption scheme, i.e. $(y'_{3,i} \cdot J_i^{-1}) \cdot (y_1'^{s})^{-1} \bmod p = W_i$. It already knows the ElGamal encryption scheme is IND-CPA secure (Tsiounis 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 $J_i = J(g^{i \cdot s \cdot r_2} \bmod p)$ with the knowledge of the public key $Y = g^s \bmod p$ and the value $y'_2 = g^{r_2} \bmod p$ 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. Theorems 4 and 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. PA$\longrightarrow$IND-CCA2 (NM-CCA2) in the random oracle model.

A PKE scheme is PA (Plaint-awareness) is for any ciphertext the adversary produces, s/he must know the corresponding the plaintext. Belleare *et al.* (1998) proved that if a PKE scheme is secure in the 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', λ)*-plaintext extractor* $\mathcal{P}$ *such that*

$$t' = t + q_J q_H (t_{\mathcal{E}} + c) \quad and \quad \lambda = 1 - \left(\frac{q_J}{2^k} + |H| \cdot \frac{1}{2^\iota}\right).$$

$t_{\mathcal{E}}$ *denotes the computational running time of the encryption algorithm* $\mathcal{E}$ *such that* $(y_1', y_3' \cdot (Y^{[[h_v]_{2k_1}]_{k_1}})^{-1} \bmod p) = \mathcal{E}_{pk}(h||H_v, [[h]_{2k_1}]^{k_1})$ *in the specification of* $\mathcal{P}$. $|H|$ *denotes the number of pairs* (h, H_v) *in the set* H_s

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

Extractor: $\mathcal{P}(hH, jJ, C, (y_1', y_2', y_3'), pk)$
For $u = 1, \ldots, q_J$ do
  For $v = 1, \ldots, q_H$ do
    $(y_1, y_3) \leftarrow (y_1', y_3' \cdot J_u^{-1} \bmod p)$
    If $(y_1, y_3) == \mathcal{E}_{pk}(h_v||H_v, [[h_v]_{2k_1}]^{k_1})$
      If $j_u == Y^{[[h_v]_{2k_1}]_{k_1}} \bmod p$
        then $x \leftarrow [h_v]^{k_0}$ and break
      Else $x \leftarrow$ 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 $t + q_J q_H (t_{\mathcal{E}} + c)$.

Since there exists an additional random oracle $J(\cdot)$, $jJ = \{(j_1, J_1), \ldots, (j_{q_J}, J_{q_J}))\}$ denotes the set of all $\mathcal{B}$'s queries and the corresponding answers of $J(\cdot)$. Intuitionally, 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 $x = \mathcal{D}_{sk}^\dagger(y_1', y_2', y_3')$. Consider the following events.

Con1$\wedge$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 $(y_1, y_3) == \mathcal{E}_{pk}(h_v||H_v, [[h_v]_{2k_1}]^{k_1})$ and $j_u == Y^{[[h_v]_{2k_1}]_{k_1}} \bmod p$ in the specification of $\mathcal{P}$ hold. Two conditions are separately denoted as Con1 and Con2.

Fail: an event assigned to be true if $x \neq \mathcal{D}^{\dagger}_{sk}(y'_1, y'_2, y'_3)$.

We now bound the failure probability as follows:

$$\begin{aligned}\Pr[\mathsf{Fail}] &= \Pr[\mathsf{Fail}|\mathsf{Con1} \wedge \mathsf{Con2}] \cdot \Pr[\mathsf{Con1} \wedge \mathsf{Con2}]\\ &\quad + \Pr[\mathsf{Fail}|\mathsf{Con1} \wedge \neg\mathsf{Con2}] \cdot \Pr[\mathsf{Con1} \wedge \neg\mathsf{Con2}]\\ &\quad + \Pr[\mathsf{Fail}|\neg\mathsf{Con1}] \cdot \Pr[\neg\mathsf{Con1}]\\ &\leqslant \Pr[\mathsf{Fail}|\mathsf{Con1} \wedge \mathsf{Con2}] + \Pr[\mathsf{Con1} \wedge \neg\mathsf{Con2}]\\ &\quad + \Pr[\mathsf{Fail}|\neg\mathsf{Con1}].\end{aligned}$$

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

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

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

$$\Pr[\mathsf{Con1} \wedge \neg\mathsf{Con2}] \leqslant \Pr[\mathsf{Con1}|\neg\mathsf{Con2}].$$

When $\neg\mathsf{Con2}$ is true, there is a J_u in the list jJ such that $(y'_1, y'_3 \cdot J_u^{-1} \bmod p) = \mathcal{E}_{pk}(h_v||H_v, [[h_v]_{2k_1}]^{k_1})$. Under the random oracle model assumption, the probability of such J_u is $\frac{1}{2^k}$. The conditional probability $\Pr[\mathsf{Con1}|\neg\mathsf{Con2}]$ is $\frac{q_J}{2^k}$.

For $\Pr[\mathsf{Fail}|\neg\mathsf{Con1}]$, $\neg\mathsf{Con1}$ 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 $J = J(Y^{[[h_v]_{2k_1}]_{k_1}} \bmod p)$, let H_s be the set of (h, H_v) such that $(y'_1, y'_3 \cdot J^{-1} \bmod p) = \mathcal{E}_{pk}(h||H_v,\ [[h]_{2k_1}]^{k_1})$. Then since $(y'_1, y'_2, y'_3) \notin C = \{(y'_1, y'_2, y'_3)_1, \ldots, (y'_1, y'_2, y'_3)_{q_E}\}$ and hence $\mathcal{D}_{sk}((y'_1, y'_3 \cdot J^{-1} \bmod p)_i) \neq h||H(h)$ 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 $\mathcal{B}$ doesn't ask query h to oracle $H(\cdot)$,

$$\Pr[\mathsf{Fail}|\neg\mathsf{Con1}] = \Pr_{H \leftarrow \Omega}\big[H(h) \in H\big] = |H| \cdot \frac{1}{2^{\iota}},$$

Obviously, $|H|$ is small. We conclude that $\Pr[\mathsf{Fail}] \leqslant \frac{q_J}{2^k} + \frac{|H|}{2^{\iota}}$. Hence, $\epsilon = 1 - \Pr[\mathsf{Fail}] = 1 - (\frac{q_J}{2^k} + \frac{|H|}{2^{\iota}})$.

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

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

Proof. We construct a breaking algorithm $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ in the IND-CPA and standard model setting by using $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ as an oracle.

Firstly, $\mathcal{A}'$ initiates two lists hH and jJ, to empty. Basically, when $\mathcal{A}$ asks query h and j, $\mathcal{A}'$ simulates two random oracles $H(\cdot)$ and $J(\cdot)$ as follows: If h has not been asked in the list hH, $\mathcal{A}'$ provides a random string H of length ι-bit, and adds an entry (h, H) to the list hH. Similarly, if j has not been asked in the list jJ, $\mathcal{A}'$ provides a random string J of length k-bit, and adds an entry (j, J) to the list jJ. When $\mathcal{A}_1$ halts and outputs (x_0, x_1, ω), $\mathcal{A}'_1$ outputs $(x_0||\gamma_0||\beta_0, x_1||\gamma_1||\beta_1, \omega)$ where γ_0, γ_1 are $(2k_1)$-bit random strings and β_0, β_1 are ι-bit random strings.

```
Adversary:  A'_1(pk)
            hH, jJ ← 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 ∉ jJ
                    J ←_R {0,1}^k
                    Put (j, J) on jJ
                    Answer J to A_1
                  Else j ∈ jJ
                    Answer J to A_1 such that (j, J) ∈ jJ
                Else if A_1 makes H-query h
                  If h ∉ hH
                    H ←_R {0,1}^ι
                    Put (h, H) on hH
                    Answer H to A_1
                  Else h ∈ hH
                    Answer H to A_1 such that (h, H) ∈ hH
              A_1 outputs (x_0, x_1, ω)
            γ_0, γ_1 ←_R {0,1}^{2k_1}
            β_0, β_1 ←_R {0,1}^ι
            Return (x_0||γ_0||β_0, x_1||γ_1||β_1, ω)
End.
```

Then, outside of $\mathcal{A}'$, the ciphertext $(y_1, y_3) = \mathcal{E}_{pk}(x_b||\gamma_b||\beta_b, R)$ is computed by the encryption oracle $\mathcal{O}_{en}$, 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 $\mathcal{A}_2$.

```
Encryption oracle:  O_en(x_0||γ_0||β_0, x_1||γ_1||β_1, pk)
                    R ←_R Z_q
                    b ←_R {0,1}
                    (y_1, y_3) ← E_pk(x_b||γ_b||β_b, R)
                    Return (y_1, y_3)
End.
```

$\mathcal{A}'_2$ chooses a random string $r_2 \in \mathbb{Z}_q$ and k-bit random string J^*. Then it sets $y'_1 = y_1$, $y'_2 = g^{r_2} \bmod p$, and $y'_3 = y_3 \cdot J^* \bmod p$. Note that (y'_1, y'_2, y'_3) is treated as the ciphertext of x_b.

Adversary: $\mathcal{A}'_2(x_0||\gamma_0||\beta_0, x_1||\gamma_1||\beta_1, \omega, (y_1, y_3))$
$r_2 \leftarrow_R \mathbb{Z}_q$; $J^* \leftarrow_R \{0,1\}^k$
$y'_1 \leftarrow y_1$; $y'_2 \leftarrow g^{r_2} \bmod p$; $y'_3 \leftarrow y_3 \cdot J^* \bmod p$
Run $\mathcal{A}_2(x_0, x_1, \omega, (y'_1, y'_2, y'_3))$
Do while $\mathcal{A}_2$ does not make H-query h and J-query j
Ask$j$ ←false
If $\mathcal{A}_2$ makes J-query j
If $j = Y^{r_2} \bmod p$
Answer J^* to $\mathcal{A}_2$
Put (j, J^*) on jJ
Ask$j$ ←true
Else if $j \notin jJ$
$J \leftarrow_R \{0,1\}^k$
Answer J to $\mathcal{A}_2$
Else $j \in jJ$
Answer J to $\mathcal{A}_2$ such that $(j, J) \in jJ$
Else if $\mathcal{A}_2$ makes H-query h
If Ask$j$ =true and $h = x_b||\gamma_b$
Stop $\mathcal{A}_2$ and output b
Else if $h \notin hH$
$H \leftarrow_R \{0,1\}^\iota$
Put (h, H) on hH
Answer H to $\mathcal{A}_2$
Else $h \in hH$
Answer H to $\mathcal{A}_2$ such that $(h, H) \in hH$
$\mathcal{A}_2$ outputs b
Return b

End.

The argument behind the proof is as follows: When $\mathcal{A}_2$ asks the query $j = Y^{r_2} \bmod p$, $\mathcal{A}'_2$ answers J^* and Ask$j$ is set be true. Since the random string r_2 is chosen by $\mathcal{A}'_2$, it has the ability to check whether the query j is equal to $Y^{r_2} \bmod p$ or not. Once Ask$j$ is true and $\mathcal{A}_2$ asks a query $h = x_b||\gamma_b$, it is almost equivalent to $\mathcal{D}_{sk}(y_1, y_3) = \mathcal{D}_{sk}(y'_1, y'_3 \cdot (J^*)^{-1} \bmod p)$, since $\mathcal{A}_2$ has no clue to $\gamma_{\bar{b}}$ where $\bar{b}$ is the complement of bit b. The probability to ask $h = x_{\bar{b}}||\gamma_{\bar{b}}$ is $\frac{1}{2^{2k_1}}$ which is negligible. Under the condition Ask$j$ is true, $\mathcal{A}'_2$ can expect that it will output a correct bit b if $\mathcal{A}_2$ asks either $h = x_0||\gamma_0$ or $h = x_1||\gamma_1$. If $\mathcal{A}_2$ asks neither of them, $\mathcal{A}'_2$ can expect that $\mathcal{A}_2$ cannot distinguish (y'_1, y'_2, y'_3) from a correct ciphertext.

To analyze the success probability of $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$, the definitions of success probabilities of $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in Definition 6 are recalled. Consider the follows events to capture the success probabilities of $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$

Ask$j$: is true if a J-query $j = Y^{r_2} \bmod p$ was made by $\mathcal{A}_2$.
Ask$b$: is true if a H-query $h = x_b||\gamma_b$ was made by $\mathcal{A}_2$.
Ask$\bar{b}$: is true if a H-query $h = x_{\bar{b}}||\gamma_{\bar{b}}$ was made by $\mathcal{A}_2$.

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

$$\begin{aligned}\Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A}',\Pi}(k)\big] &= \Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A}',\Pi}(k)|\mathsf{Ask}j \wedge \mathsf{Ask}b\big]\cdot\Pr\big[\mathsf{Ask}j \wedge \mathsf{Ask}b\big]\\ &+ \Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A}',\Pi}(k)|\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b\big]\\ &\times \Pr[\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b].\end{aligned}$$

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

$$\begin{aligned}\Pr[\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b] &= \Pr\big[(\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \mathsf{Ask}\bar{b}\big]\\ &+ \Pr\big[(\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \neg\mathsf{Ask}\bar{b}\big].\end{aligned}$$

Then,

$$\begin{aligned}\Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A}',\Pi}(k)\big] &= \Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A}',\Pi}(k)|\mathsf{Ask}j \wedge \mathsf{Ask}b\big]\cdot\Pr[\mathsf{Ask}j \wedge \mathsf{Ask}b]\\ &+ \Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A}',\Pi}(k)|(\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \mathsf{Ask}\bar{b}\big]\\ &\times \Pr\big[(\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \mathsf{Ask}\bar{b}\big]\\ &+ \Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A}',\Pi}(k)|(\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \neg\mathsf{Ask}\bar{b}\big]\\ &\times \Pr\big[(\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \neg\mathsf{Ask}\bar{b}\big].\end{aligned}$$

Similarly,

$$\begin{aligned}\Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A},\Pi^\dagger}(k)\big] &= \Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A},\Pi^\dagger}(k)|\mathsf{Ask}j \wedge \mathsf{Ask}b\big]\cdot\Pr[\mathsf{Ask}j \wedge \mathsf{Ask}b]\\ &+ \Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A},\Pi^\dagger}(k)|(\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \mathsf{Ask}\bar{b}\big]\\ &\times \Pr\big[(\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \mathsf{Ask}\bar{b}\big]\\ &+ \Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A},\Pi^\dagger}(k)|(\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \neg\mathsf{Ask}\bar{b}\big]\\ &\times \Pr\big[(\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \neg\mathsf{Ask}\bar{b}\big].\end{aligned}$$

From the specification of $\mathcal{A}'$, we have the following equations,

$$\begin{cases}\Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A},\Pi^\dagger}(k)|\mathsf{Ask}j \wedge \mathsf{Ask}b\big] = 1,\\ \Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A},\Pi^\dagger}(k)|(\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \mathsf{Ask}\bar{b}\big] = 0\\ \Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A},\Pi^\dagger}(k)|(\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \neg\mathsf{Ask}\bar{b}\big]\\ \quad = \Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A}',\Pi}(k)|(\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \neg\mathsf{Ask}\bar{b}\big]\end{cases}$$

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

$$\begin{aligned}&\Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A}',\Pi}(k)\big] - \Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A},\Pi^\dagger}(k)\big]\\ &\quad = \big(1 - \Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A},\Pi^\dagger}(k)|\mathsf{Ask}j \wedge \mathsf{Ask}b\big]\big)\cdot\Pr[\mathsf{Ask}j \wedge \mathsf{Ask}b]\end{aligned}$$

$$- \Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A},\Pi^\dagger}(k) | (\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \mathsf{Ask}\bar{b}\big] \cdot \Pr\big[(\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \mathsf{Ask}\bar{b}\big]$$
$$\geqslant -\Pr\big[(\neg\mathsf{Ask}j \vee \neg\mathsf{Ask}b) \wedge \mathsf{Ask}\bar{b}\big]$$
$$= -\Pr\big[(\neg\mathsf{Ask}j \wedge \mathsf{Ask}\bar{b}) \vee (\neg\mathsf{Ask}b) \wedge \mathsf{Ask}\bar{b})\big]$$
$$\geqslant -\big(\Pr[\neg\mathsf{Ask}j \wedge \mathsf{Ask}\bar{b}] + \Pr[\neg\mathsf{Ask}b \wedge \mathsf{Ask}\bar{b}]\big).$$

Since $\gamma_{\bar{b}}$ is a uniform random string over $\{0,1\}^{2k_1}$, we have $\Pr[\neg\mathsf{Ask}j \wedge \mathsf{Ask}\bar{b}] \leqslant \frac{q_H}{2^{2k_1}}$ and $\Pr[\neg\mathsf{Ask}b \wedge \mathsf{Ask}\bar{b}] \leqslant \frac{q_H}{2^{2k_1}}$. Thus,

$$\Pr\big[\mathsf{Succ}^{\mathsf{IND-CPA}}_{\mathcal{A},\mathsf{PKE}}(k)\big] - \big(\Pr\big[\neg\mathsf{Ask}j \wedge \mathsf{Ask}\bar{b}\big] + \Pr\big[\neg\mathsf{Ask}b \wedge \mathsf{Ask}\bar{b}\big]\big)$$
$$\geqslant \frac{\epsilon+1}{2} - \frac{q_H}{2^{2k_1-1}}.$$

and we obtain that $\epsilon' = \epsilon - \frac{q_H}{2^{(2k_1-2)}}$. The running time of $\mathcal{A}'$ is at most time $t + q_H \cdot c + q_J \cdot c$.

Theorem 6. PKE: IND-CCA2. *If there exists a $(t, q_H, q_J, q_D, \epsilon)$-breaker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ for $\Pi^\dagger = (\mathcal{K}^\dagger, \mathcal{E}^\dagger, \mathcal{D}^\dagger)$ in the sense of* IND-CCA2 *in the random oracle model, then there exist a constant c and a (t', ϵ')-breaker $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ for $\Pi = (\widehat{\mathcal{K}}, \mathcal{E}, \mathcal{D})$ in the sense of* IND-CPA *in the standard model where*

$$t' = t + q_H q_J (t_{\mathcal{E}} + c) + q_H c + q_J c \quad \textit{and} \quad \epsilon' = (\epsilon - \frac{q_H}{2^{(2k_1-2)}}) \cdot \lambda^{q_D}.$$

Proof. From the result of Theorem 6, it is found out that the encryption scheme $\Pi^\dagger$ is secure in the IND-CCA2. The proof is omitted since it is clear from the following specification of adversary $\mathcal{A}'$ combined with the proofs in Theorems 4 and 5.

Adversary: $\mathcal{A}'_1(pk)$
$hH, jJ \leftarrow$empty
Run $\mathcal{A}_1^{\mathcal{D}_{sk},H,J}(pk)$
  Do while $\mathcal{A}_1$ does not make H-query h, J-query j, D-query $(y_1, y_2, y_3)'$
    If $\mathcal{A}_1$ makes J-query j
      If $j \notin jJ$
        $J \leftarrow_R \{0,1\}^k$
        Put (j, J) on jJ
        Answer J to $\mathcal{A}_1$
      Else $j \in jJ$
        Answer J to $\mathcal{A}_1$ such that $(j, J) \in jJ$
    Else if $\mathcal{A}_1$ makes H-query h
      If $h \notin hH$
        $H \leftarrow_R \{0,1\}^\iota$
        Put (h, H) on hH
        Answer H to $\mathcal{A}_1$

Else $h \in hH$
Answer H to $\mathcal{A}_1$ such that $(h, H) \in hH$
Else if $\mathcal{A}_1$ makes D-query $(y_1, y_2, y_3)'$
Run $\mathcal{P}(hH, jJ, C, (y_1, y_2, y_3)', pk)$
$\mathcal{P}$ outputs x'
Answer x' to $\mathcal{A}_1$
$\mathcal{A}_1$ outputs (x_0, x_1, ω)
$\gamma_0, \gamma_1 \leftarrow_R \{0,1\}^{2k_1}$
$\beta_0, \beta_1 \leftarrow_R \{0,1\}^{\iota}$
Return $(x_0||\gamma_0||\beta_0, x_1||\gamma_1||\beta_1, \omega)$
End.

Encryption oracle: $\mathcal{O}_{en}(x_0||\gamma_0||\beta_0, x_1||\gamma_1||\beta_1, pk)$
$b \leftarrow_R \{0,1\}$
$(y_1, y_3) \leftarrow \mathcal{E}_{pk}(x_b||\gamma_b||\beta_b, R)$
Return (y_1, y_3)
End.

Adversary: $\mathcal{A}_2'(x_0||\gamma_0||\beta_0, x_1||\gamma_1||\beta_1, \omega, (y_1, y_3))$
$r_2 \leftarrow_R \mathbb{Z}_q$; $J^* \leftarrow_R \{0,1\}^k$
$y_1' \leftarrow y_1$; $y_2' \leftarrow g^{r_2} \bmod p$; $y_3' \leftarrow y_3 \cdot J^* \bmod p$
Run $\mathcal{A}_2^{\mathcal{D}_{sk}, H, J}(x_0, x_1, \omega, (y_1', y_2', y_3'))$
$C \leftarrow (y_1', y_2', y_3')$
Do while $\mathcal{A}_2$ does not make H-query h and J-query j
D-query $(y_1, y_2, y_3)'$
Ask$j$ $\leftarrow$false
If $\mathcal{A}_2$ makes J-query j
If $j = Y^{r_2} \bmod p$
Answer J^* to $\mathcal{A}_2$
Put (j, J^*) on jJ
Ask$j$ $\leftarrow$true
Else if $j \notin jJ$
$J \leftarrow_R \{0,1\}^k$
Answer J to $\mathcal{A}_2$
Else $j \in jJ$
Answer J to $\mathcal{A}_2$ such that $(j, J) \in jJ$
Else if $\mathcal{A}_2$ makes H-query h
If Ask$j$ =true and $h = x_b||\gamma_b$
Stop $\mathcal{A}_2$ and output b
Else if $h \notin hH$
$H \leftarrow_R \{0,1\}^{\iota}$
Put (h, H) on hH
Answer H to $\mathcal{A}_2$

Else $h \in hH$

Answer H to $\mathcal{A}_2$ such that $(h, H) \in hH$

Else if $\mathcal{A}_2$ makes D-query $(y_1, y_2, y_3)'$

Run $\mathcal{P}(hH, jJ, C, (y_1, y_2, y_3)', pk)$

$\mathcal{P}$ outputs x'

Answer x' to $\mathcal{A}_1$

$\mathcal{A}_2$ 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 $x_1, x_2, \ldots, x_n$ 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 $\frac{n \cdot k}{k_0} = n + \lceil \frac{n \cdot (2k_1 + \iota)}{k_0} \rceil$. Let $n' = n + \lceil \frac{n \cdot (2k_1 + \iota)}{k_0} \rceil$. 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 $x_2, \ldots, x_{n'}$, it is not necessary to compute the values $y_1 = g^{r_1} \bmod p$, $y_2 = g^{r_2} \bmod p$, $Y^{r_1} \bmod p$, and $Y^{r_2} \bmod p$ 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 $(y_{3,2}, \ldots, y_{3,n'})$, the values $y_1'^s \bmod p$ and $y_2'^s \bmod p$ 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 in 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 $\mathcal{K}'$ or $\widehat{\mathcal{K}}$). Besides, the cryptosystem has the probability to be crashed when $Y^{r_1} \oplus (Y^{r_2})^i \bmod p = 0$. 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 $g^a \bmod p$ (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

Abe, M., Gennaro, R., Kurosawa, K., Shoup, V. (2005). Tag-KEM/DEM: a new framewrok for hybrid encryption and a new analysis of Kurosawa–Desmedt KEM. In: *Advances in Cryptology (EUROCRYPT'05), Lecture Notes in Computer Science*, Vol. 3494, pp. 128–146.

Bao, F., Lee, C.C., Hwang, M.S. (2006). Cryptanalysis and improvement on batch verifying multiple RSA digital signatures. *Applied Mathematics and Computation*, 172(2), 1195–1200.

Bellare, M., Rogaway, P. (1993). Random oracles are practical: a paradigm for designing efficient protocols. In: *1st Annual Conference on Computer and Communications Security*, ACM, pp. 62–73.

Bellare, M., Rogaway, P. (1994). Optimal asymmetric encryption. In: *Advances in Cryptology (EUROCRYPT'94), Lecture Notes in Computer Science*, Vol. 950, pp. 92–111.

Bellare, M., Desai, A., Pointcheval, D., Rogaway, P. (1998). Relations among notations of security for public-key encryption schemes. In: *Advances in Cryptology (CRYPTO'98), Lecture Notes in Computer Science*, Vol. 1462, pp. 26–45.

Chang, T.Y. (2008). A convertible multi-authenticated encryption scheme for group communications. *Information Sciences*, 178(17), 3426–3434.

Chang, T.Y. (2009). An id-based group-oriented decryption scheme secure against adaptive chosen-ciphertext attacks. *Computer Communications*, 32(17), 1829–1836.

Chang, T.Y. (2010). An computation-efficient generalized group-oriented cryptosystem. *Informatica*, 21(3), 1–14.

Chmielowiec, A. (2010). Fixed points of the RSA encryption algorithm. *Theoretical Computer Science*, 411(1), 288–292.

Cramer, R., Shoup, V. (1998). A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: *Advances in Cryptology (CRYPTO'98), Lecture Notes in Computer Science*, Vol. 1462, pp. 13–25.

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

Fujisaki, E., Okamoto, T., Pointcheval, D., Stern, J. (2001). RSA-OAEP is secure under the RSA assumption. In: *Advances in Cryptology (CRYPTO'01), Lecture Notes in Computer Science*, Vol. 2139, pp. 260–274.

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

Hwang, M.S., Hwang, K.F., Lin, I.C. (2000). Cryptanalysis of the batch verifying multiple RSA digital signatures. *Informatica*, 11(1), 1–4.

Hwang, M.S., Chang, C.C., Hwang, K.F. (2002). An ElGamal-like cryptosystem for enciphering large messages. *IEEE Transactions on Knowledge and Data Engineering*, 14(2), 445–446.

Hwang, M.S., Lu, E.J.L., Lin, I.C. (2003). A practical (t, n) threshold proxy signature scheme based on the RSA cryptosystem. *IEEE Transactions on Knowledge and Data Engineering*, 15(6), 1552–1560.

Lee, C.C., Hwang, M.S., Tzeng, S.F. (2009). A new convertible authenticated encryption scheme based on the ElGamal cryptosystem. *International Journal of Foundations of Computer Science*, 20(2), 351–359.

Naor, M., Yung, M. (1990). Public-key cryptosystems provably secure against chosen ciphertext attack. In: *Proc. of the 22st STOC*, pp. 427–43.

Paillier, P., Pointcheval, D. (1999). Efficient public-key cryptosystems provaly secure against active adversaries. In: *Advances in Cryptology (ASIACRYPT'99), Lecture Notes in Computer Science*, Vol. 1716, pp. 165–179.

Pointcheval, D. (1999). New public key cryptosystems based on the dependent-RSA problems. In: *Advances in Cryptology (EUROCRYPT'99), Lecture Notes in Computer Science*, Vol. 1592, pp. 239–254.

Rackoff, C., Simon, D. (1991). Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: *Advances in Cryptology (CRYPTO'91), Lecture Notes in Computer Science*, Vol. 576, pp. 433–444.

Rivest, R.L., Shamir, A., Adleman, L. (1978). A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 21, 120–126.

Shen, J.J., Lin, C.W., Hwang, M.S. (2003). A modified remote user authentication scheme using smart cards. *IEEE Transactions on Consumer Electronics*, 49(2), 414–416.

Shoup, V. (1997). Lower bounds for discrete logairhms and related problems. In: *Advances in Cryptology (EUROCRYPTO'97), Lecture Notes in Computer Science*, Vol. 1233, pp. 256–266.

Shoup, V. (2001). OAEP reconsidered. In: *Advances in Cryptology (CRYPTO'01), Lecture Notes in Computer Science*, Vol. 2139, pp. 239–259.

Shoup, V., Gennaro, R. (1998). Securing threhshold cryptosystem against chosen ciphertext attack. In: *Advances in Cryptology (EUROCRYPT'98), Lecture Notes in Computer Science*, Vol. 1403, pp. 1–16.

Tsiounis, Y., Yung, M. (1998). On the security of ElGamal based encryption. In *PKC'98*, pp. 117–134.

Wang, B., Hu, Y. (2010). A Novel Combinatorial Public Key Cryptosystem. *Informatica*, 21(4), 611–626.

Wang, M.N., Yen, S.M., Wu, C.D., Lin, C.T. (2006). Cryptanalysis on an ElGamal-like cryptosystem for encrypting large messages. In: *Proceedings of the 6th ESEAS International Conference on Applied Informatics and Communications*, pp. 418–422.

Yang, C.C., Chang, T.Y., Li, J.W., Hwang, M.S. (2003). Simple generalized group-oriented cryptosystems using ElGamal cryptosystem. *Informatica*, 14(1), 111-120.

T.Y. Chang received his MS from the Graduate Institute of Computer Science and Information Engineering from the Chaoyang University of Technology in 2003, and a PhD 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.

M.-S. Hwang received the BS in electronic engineering from the National Taipei Institute of Technology, Taipei, Taiwan, Republic of China (ROC), in 1980; the MS 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.

W.-P. Yang was born on May 17, 1950 in Hualien, TAIWAN. He received the BS degree in mathematics from National Taiwan Normal University in 1974, and the MS and PhD 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, California 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 miming, 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 MS Thesis Supervision, 1993 AceR Long Term Award for Outstanding PhD 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.

Patobulintos ElGamal'io viešojo rakto šifravimo schemos, skirtos didelės apimties pranešimams šifruoti, kriptoanalizė

Ting-Yi CHANG, Min-Shiang HWANG, Wei-Pang YANG

Hwang ir kt. pasiūlė ElGamal'io tipo schemą, skirtą didelės apimties pranešimams šifruoti, kuri yra efektyvesnė skaičiavimo sudėtingumo ir duomenų transformacijų kiekio prasmėmis. Jie teigė, kad schema yra saugi pasirinkto atvirojo teksto atakoms esant prielaidai, kad Diffie-Helman'o problema yra neišsprendžiama. Vėliau Wang ir kt. parodė, kad Hwang'o schemos sauga nėra pakankama ir galimi nesėkmingo dešifravimo atvejai. Be to jie patobulino Hwang ir kt. schemą padidindami jos saugumą ir sumažindami nesėkmingo dešifravimo galimybę. Šiame straipsnyje parodyta, kad jų schema yra vis dar nesaugi nuo pasirinkto teksto atakų. Taip pat pasiūlyta nauja ElGamal'io tipo schema, atspari pasirinkto teksto atakoms.