How to Enhance the Security of Public-Key Encryption at Minimum Cost

Eiichiro Fujisaki and Tatsuaki Okamoto

NTT Laboratories,
1-1 Hikarinooka, Yokosuka-shi, 239-0847 Japan
{fujisaki, okamoto}@sucaba.isl.ntt.co.jp

Abstract. This paper presents a simple and efficient conversion from a semantically secure public-key encryption scheme against *passive adversaries* to a non-malleable (or semantically secure) public-key encryption scheme against *adaptive chosen-ciphertext attacks* (*active adversaries*) in the random oracle model. Since our conversion requires only one random (hash) function operation, the converted scheme is almost as efficient as the original one, when the random function is replaced by a practical hash function such as SHA-1 and MD5. We also give a concrete analysis of the reduction for proving its security, and show that our security reduction is (almost) optimally efficient. Finally this paper gives some practical examples of applying this conversion to some practical and semantically secure encryption schemes such as the ElGamal, Blum-Goldwasser and Okamoto-Uchiyama schemes [4,7,9].

1 Introduction

1.1 Background

One of the most important topics in cryptography is to propose a practical and provably secure public-key encryption scheme. The strongest security notion in the public-key encryption is that of non-malleability or semantical security against adaptive chosen-ciphertext attacks. In [3], Bellare, Desai, Pointcheval and Rogaway show that semantical security against adaptive chosen-ciphertext attacks (IND-CCA2) is equivalent to (or sufficient for) the strongest security notion (NM-CCA2).

A promising way to construct a practical public-key encryption scheme semantically secure against adaptive chosen-ciphertext attacks (IND-CCA2) is to convert from a primitive trap-door one-way function (such as RSA or ElGamal) by using *random functions*. Here, an ideally random function, the "random oracle", is assumed when proving the security, and the random function is replaced by a practical random-like function such as a one-way hash function (e.g., SHA-1 and MD5, etc.) when realizing it in practice. This approach was initiated by Bellare and Rogaway, and is called the *random oracle model* [2].

Although security in the random oracle model cannot be guaranteed formally when a practical random-like function is used in place of the random oracle, this

H. Imai and Y. Zheng (Eds.): PKC'99, LNCS 1560, pp. 53–68, 1999.

paradigm often yields much more efficient schemes than those in the *standard model* and gives an informal security guarantee of the schemes.

Two typical primitives of the trap-door one-way function are RSA and ElGamal. The RSA function is a trap-door one-way permutation, and the ElGamal function is a probabilistic trap-door one-way function.

Bellare and Rogaway presented a generic and efficient way to convert a trap-door one-way permutation to an IND-CCA2 secure scheme in the random oracle model (The scheme created in this way from the RSA function is called OAEP).

However, their method cannot be applied to a probabilistic trap-door one-way function such as ElGamal. Therefore, a new measure to convert a probabilistic trap-door one-way function to an IND-CCA2 secure scheme (in the random oracle model) should be very valuable.

This paper will present such a generic and efficient measure. It converts a probabilistic trap-door one-way function to an IND-CCA2 secure scheme in the random oracle model provided that the trap-door one-way function is semantically secure (IND-CPA).

Since our conversion requires only one random (hash) function operation, the converted scheme is almost as efficient as the original scheme, when the random function is replaced by a practical hash function such as SHA-1 and MD5. Therefore, we can construct practical IND-CCA2 secure schemes (in the random oracle model) based on several practical IND-CPA secure schemes (under some reasonable assumptions) such as the (elliptic curve) ElGamal, Blum-Goldwasser and Okamoto-Uchiyama schemes [4,7,9,11].

We begin by examining the notions of public-key encryption security.

1.2 Classification of Encryption Scheme Security

We can define the security levels of public-key encryption schemes, using the pairs of *goals* and *adversary models* (We saw this classification first in the paper of [3], which stated that the viewpoint was suggested to the authors by Naor).

The goals are one-wayness (OW), indistinguishability (IND) [8],and non-malleability (NM) [6] of encryption. One-wayness (OW) is defined by the adversary's inability, given a challenge ciphertext y, to decrypt y and get the whole plaintext x. Indistinguishability (IND) is defined by the adversary's inability, given a challenge ciphertext y, to learn any information about the plaintext x. Non-malleability (NM) is defined by the adversary's inability, given a challenge ciphertext y, to get a different ciphertext y' such that the corresponding plaintexts, x and x', are *meaningfully* related. Here a *meaningful* relation is, for instance, $x = x' + 1$.

The three adversary models are called chosen plaintext attack model (CPA), non-adaptive chosen-ciphertext attack model (CCA1), and adaptive chosen ciphertext attack model (CCA2). In CPA, the adversary is given only the public key. Of course, she can get the ciphertext of any plaintext chosen by her. Clearly, in public-key encryption schemes, this attack cannot be avoided. In CCA1, in addition to the public key, the adversary can access to the decryption oracle

although she is only allowed to access to the oracle before given a challenge ciphertext. In CCA2, the adversary can access to the decryption oracle anytime (before or after given a challenge ciphertext). She is only prohibited from asking for the decryption of the challenge ciphertext itself.

Furthermore, we separate public-key encryption schemes into the random oracle (RO) model or the standard model. In the random oracle model, every adversary, independent of the adversary models, can be allowed to access to the random oracle anytime,

We say, for the security of public-key encryption scheme Π, that Π is secure in the sense of GOAL-ATK in the RO (or standard) model, where GOAL = {OW, IND, NM} and ATK = {CPA, CCA1,CCA2}. Here one can think of pairs of *goals* and *attacks*; OW-CPA, ..., OW-CCA2, IND-CPA, ..., NM-CCA2. According to [3], the relations among each notion of security are as follows: [1]

$$\begin{array}{ccccc}
\text{NM-CPA} & \longleftarrow & \text{NM-CCA1} & \longleftarrow & \text{NM-CCA2} \\
 & & & \not\longleftarrow & \\
\downarrow & \searrow\!\!\!\nearrow & \downarrow & & \downarrow\!\!\uparrow \\
\text{IND-CPA} & \longleftarrow & \text{IND-CCA1} & \longleftarrow & \text{IND-CCA2} \\
\downarrow & & \downarrow & & \downarrow \\
\text{OW-CPA} & \longleftarrow & \text{OW-CCA1} & \longleftarrow & \text{OW-CCA2}
\end{array}$$

Here, for $\mathbb{A}, \mathbb{B} \in$ GOAL-ATK "$\mathbb{A} \rightarrow \mathbb{B}$" (say, $\mathbb{A}$ implies $\mathbb{B}$) denotes that encryption scheme $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ being secure in the sense of $\mathbb{A}$ is also secure in the sense of $\mathbb{B}$, while "$\mathbb{A} \not\rightarrow \mathbb{B}$" (say, $\mathbb{A}$ doesn't imply $\mathbb{B}$) denotes Π being secure in the sense of $\mathbb{A}$ is not always secure in the sense of $\mathbb{B}$.

We will provide precise definitions of these notations in Sec.2 (Due to the space limitation, one-wayness is not discussed).

1.3 Our Results

This paper shows a simple and efficient conversion from an IND-CPA secure public-key encryption scheme to an NM-CCA2 (or IND-CCA2) secure public-key encryption scheme in the random oracle model.

Suppose $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is an IND-CPA secure public-key encryption scheme and $\mathcal{E}_{pk}(X, R)$ is encryption function in it, where pk is a public-key, X is a message with $k + k_0$ bits and R is a random string with l bits. The conversion is

$$\bar{\mathcal{E}}_{pk}(x, r) := \mathcal{E}_{pk}(x||r, H(x||r)), \tag{1}$$

where H is a random function of $\{0,1\}^{k+k_0} \longrightarrow \{0,1\}^l$, x is a message of the converted public-key encryption scheme $\bar{\Pi} := (\bar{\mathcal{K}}, \bar{\mathcal{E}}, \bar{\mathcal{D}})$, r is a random string with k_0 bits, and $||$ denotes concatenation.

[1] Although one-wayness is not described in [3], the relations among OW and other goals in the diagram are clear.

Main Theorem (Theorem 3)
Suppose that $\Pi(1^{k+k_0})$ *is the original IND-CPA secure scheme and* $\bar{\Pi}$ *is the converted scheme. If there exists a* (t, q_H, q_D, ϵ)*-breaker* A *for* $\bar{\Pi}(1^k)$ *in the sense of IND-CCA2 in the random oracle model, then there exist constants,* c *and* $(t', 0, 0, \epsilon')$*-breaker* A' *for* $\Pi(1^{k+k_0})$ *where*

$$t' = t + q_H \cdot (T_{\mathcal{E}}(k) + c \cdot k), \text{ and}$$
$$\epsilon' = (\epsilon - q_H \cdot 2^{-(k_0-1)}) \cdot (1 - 2^{-l_0})^{q_D}.$$

Here, (t, q_H, q_D, ϵ)*-breaker* A*, informally, means that* A *stops within* t *steps, succeeds with probability* $\geq \epsilon$*, makes at most* q_H *queries to random oracle* H*, and makes at most* $q_{\mathcal{D}}$ *queries to decryption oracle* $\mathcal{D}_{sk}$ *(see Sec. 2 for the formal definition).* $T_{\mathcal{E}}(k)$ *denotes the computational time of the encryption algorithm* $\mathcal{E}_{pk}(\cdot)$*, and* c_0 *and* c_1 *depend on details of the underlying model of computation.*

This theorem implies that if the original scheme Π is IND-CPA secure, the converted scheme $\bar{\Pi}$ is IND-CCA2 secure (and NM-CCA2 secure as well) in the random oracle model, provided that k, k_0 and l are in proportion to system size.

1.4 Merits and Related Works

As mentioned above, Bellare-Rogaway conversion [2] is a generic scheme to be applied to trap-door one-way permutations (such as RSA) while our conversion is a generic one to be applied to probabilistic trap-door one-way functions (such as ElGamal).

Since our conversion starts from an IND-CPA secure scheme, which is more secure than Bellare-Rogaway conversion does, our conversion is simpler and more efficient than theirs, i.e., our conversion requires only one random function operation, while Bellare-Rogaway conversion requires two random function operations. In addition, the security reduction of our conversion is more efficient (tight) than that of Bellare-Rogaway's, since we need no additional reduction for semantical security.

Recently, Cramer and Shoup presented a new public-key encryption scheme based on the ElGamal, which is the first practical IND-CCA2 secure scheme in the standard model [5]. Compared with theirs, our converted version of the ElGamal scheme has a disadvantage in terms of the assumptions (ours in the random oracle model and under the decision Diffie-Hellman assumption, while the Cramer-Shoup scheme under the universal one-way hash assumption and the decision Diffie-Hellman assumption), but ours still has better efficiency, at least twice that of theirs. In addition, since our approach is generic, unlike the Cramer-Shoup scheme, it can be adopted by other IND-CPA secure schemes such as Blum-Goldwasser and Okamoto-Uchiyama schemes [4,9].

Compared with the converted ElGamal scheme presented by Tsiounis and Yung [11], which is secure in the IND-CCA2 (i.e. NM-CCA2) sense, our converted one is at least twice as efficient as theirs under the same assumptions, the random oracle model and the decision Diffie-Hellman assumption.

2 Definitions and Security Models

In this section, we give some definitions about encryption scheme security. Basically, we follow the terminology in [2,3].

Definition 1. *Let A be a probabilistic algorithm and let $A(x_1, \ldots, x_n; r)$ be the result of A on input $(x_1, \ldots, x_n)$ and coins r. We define by $y \leftarrow A(x_1, \ldots, x_n)$ the experiment of picking r at random and letting y be $A(x_1, \ldots, x_n; r)$. If S is a finite set, let $y \leftarrow_R S$ be the operation of picking y at random and uniformly from finite set S. ε denote the null symbol and, for list τ, $\tau \leftarrow \varepsilon$ denote the operation of letting list τ be empty. Moreover, let $||$ denote the concatenation operator and, for n-bit string x, $[x]^k$ and $[x]_k$ denote the first and last k-bit strings of x respectively ($k \leq n$).*

Definition 2. **[Public-Key Encryption]** *We say that a triple of algorithm $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is a public-key encryption scheme if*

- *$\mathcal{K}$, the key-generation algorithm, is a probabilistic algorithm which on input 1^k ($k \in \mathbb{N}$) outputs, in polynomial-time in k, a pair (pk, sk) of matching public and secret keys.*
- *$\mathcal{E}$, the encryption algorithm, is a probabilistic algorithm which on input public-key pk and plaintext $x \in \{0,1\}^k$ outputs ciphertext y in polynomial-time in k. We denote by $\mathcal{E}_{pk} : \{0,1\}^k \times \{0,1\}^{l(k)} \rightarrow \{0,1\}^{n(k)}$ the map from the product of k-bit message and $l(k)$-bit coin-flipping spaces to $n(k)$-bit cipher space, where functions, $l(\cdot)$ and $n(\cdot)$, are positive integer valued functions bounded in some polynomial, namely $l(k), n(k) < \exists poly(k)$ for enough large k.*
- *$\mathcal{D}$, the decryption algorithm, is a deterministic algorithm which on input secret-key sk and ciphertext y outputs $\mathcal{D}_{sk}(y)$ such that*

$$\mathcal{D}_{sk}(y) := \begin{cases} x \in \{0,1\}^k \text{ if there exists } x \text{ such that } y = \mathcal{E}_{pk}(x) \\ \varepsilon \text{ (null)} \qquad \text{otherwise.} \end{cases}$$

We say that ciphertext y is valid if there exists a plaintext x such that $y = \mathcal{E}_{pk}(x)$. We insist that in a public-key encryption scheme the map from the plaintext space to the ciphertext space should be one-to-one (injective): the decryption of each ciphertext should be unique.

Definition 3. **[Random Oracle Model]** *We define by Ω the set of all maps from the set $\{0,1\}^*$ of finite strings to the set $\{0,1\}^\infty$ of infinite strings. $H \leftarrow \Omega$ means that we chose map H from a set of an appropriate finite length (say $: \{0,1\}^a$) to a set of an appropriate finite length (say $\{0,1\}^b$), from Ω at random and uniformly, restricting the domain to $\{0,1\}^a$ and the range to the first b bits of output. If $\mathcal{E}$ and $\mathcal{D}$ in public-key encryption scheme Π are allowed to access such identical map H, we say that the scheme is defined in the random oracle model. If we insist on the fact, then we will denote $\Pi := (\mathcal{K}, \mathcal{E}^H, \mathcal{D}^H)$.*

Below, we give the precise definitions of GOAL-ATK described in Sec.1.2. Due to the space limitations, one-wayness is not described.

Definition 4. [IND-ATK] *Let $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme and let $A := (A_1, A_2)$ be a pair of probabilistic algorithms (say Adversary). For atk $\in$ {cpa,cca1,cca2} and $k \in \mathbb{N}$, let define*

$$Adv^{ind\text{-}atk}_{A,\Pi}(k) := 2\Pr[H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}(1^k); (x_0, x_1, s) \leftarrow A_1^{O_1,H}(pk); \\ b \leftarrow_R \{0,1\}; y \leftarrow \mathcal{E}_{pk}(x_b) : A_2^{O_2,H}(x_0, x_1, s, y) = b] - 1.$$

Here, $O_1(\cdot)$, $O_2(\cdot)$ are defined as follows:

- *If atk=cpa then $O_1(\cdot) = \varepsilon$ and $O_2(\cdot) = \varepsilon$*
- *If atk=cca1 then $O_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ and $O_2(\cdot) = \varepsilon$*
- *If atk=cca2 then $O_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ and $O_2(\cdot) = \mathcal{D}_{sk}(\cdot)$*

In addition we define that A_1 outputs x_0, x_1 with $|x_0| = |x_1|$ and, in the case of IND-CCA2, A_2 does not ask its oracle to decrypt y.

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

We insist that $A := (A_1, A_2)$ is not allowed to access to H in the standard model. When we insist on that, we write $A_1^{O_1}$ and $A_2^{O_2}$ instead of $A_1^{O_1,H}$ and $A_2^{O_2,H}$, respectively. On the other hand, when we insist on the random oracle model, we write $\mathcal{E}^H_{pk}(\cdot)$ and $\mathcal{D}^H_{sk}(\cdot)$ instead of $\mathcal{E}_{pk}(\cdot)$ and $\mathcal{D}_{sk}(\cdot)$, respectively.

Definition 5. [NM-ATK] *Let $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme and let $A := (A_1, A_2)$ be a pair of probabilistic algorithms (say Adversary). For atk $\in$ {cpa,cca1,cca2} and $k \in \mathbb{N}$, let define*

$$Adv^{nm\text{-}atk}_{A,\Pi}(k) := |Succ^{nm\text{-}atk}_{A,\Pi}(k) - Succ^{nm\text{-}atk}_{A,\Pi,\$}(k)|$$

where $Succ^{nm\text{-}atk}_{A,\Pi}(k) :=$

$$\Pr[H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}(1^k); (M, s) \leftarrow A_1^{O_1,H}(pk); x, x' \leftarrow M; y \leftarrow \mathcal{E}_{pk}(x); \\ (R, \boldsymbol{y}) \leftarrow A_2^{O_2,H}(M, s, y); \boldsymbol{x} \leftarrow \mathcal{D}_{sk}(\boldsymbol{y}) : (y \notin \boldsymbol{y}) \wedge (\varepsilon(null) \notin \boldsymbol{x}) \wedge R(x, \boldsymbol{x})]$$

and $Succ^{nm\text{-}atk}_{A,\Pi,\$}(k) :=$

$$\Pr[H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}(1^k); (M, s) \leftarrow A_1^{O_1,H}(pk); x, x' \leftarrow M; y \leftarrow \mathcal{E}_{pk}(x); \\ (R, \boldsymbol{y}) \leftarrow A_2^{O_2,H}(M, s, y); \boldsymbol{x} \leftarrow \mathcal{D}_{sk}(\boldsymbol{y}) : (y \notin \boldsymbol{y}) \wedge (\varepsilon(null) \notin \boldsymbol{x}) \wedge R(x', \boldsymbol{x})]$$

Here, $O_1(\cdot)$, $O_2(\cdot)$ are defined as before. In the case of IND-CCA2, A_2 does not ask its oracle to decrypt y.

We say that M is valid if $|x| = |x'|$ for any x, x' that are given non-zero probability in the message space M.

We say that Π is secure in the sense of NM-ATK if any adversary A being polynomial-time in k outputs a valid message space M samplable in polynomial in k and a relation R computable in polynomial in k, then $Adv^{nm\text{-}atk}_{A,\Pi}(k)$ is negligible in k.

We insist that $A := (A_1, A_2)$ is not allowed to access to H in the standard model. When we insist on that, we write $A_1^{O_1}$ and $A_2^{O_2}$ instead of $A_1^{O_1,H}$ and $A_2^{O_2,H}$, respectively. On the other hand, when we insist on the random oracle model, we write $\mathcal{E}^H_{pk}(\cdot)$ and $\mathcal{D}^H_{sk}(\cdot)$ instead of $\mathcal{E}_{pk}(\cdot)$ and $\mathcal{D}_{sk}(\cdot)$, respectively.

We review some important results proven in [3] below. Here, as mentioned above, for $\mathbb{A}, \mathbb{B} \in$ GOAL-ATK "$\mathbb{A} \rightarrow \mathbb{B}$" (say, $\mathbb{A}$ implies $\mathbb{B}$) denotes that encryption scheme $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ being secure in the sense of $\mathbb{A}$ is also secure in the sense of $\mathbb{B}$, while "$\mathbb{A} \not\rightarrow \mathbb{B}$" (say, $\mathbb{A}$ doesn't imply $\mathbb{B}$) denotes Π being secure in the sense of $\mathbb{A}$ is not always secure in the sense of $\mathbb{B}$.

Proposition 1. *IND-CCA2 $\rightarrow$ NM-CCA2.*

From this proposition, it is clear that

Corollary 1. *IND-CCA2 $\longleftrightarrow$ NM-CCA2.*

Proposition 2. *IND-CCA1 $\not\rightarrow$ NM-CCA2.*

The following definition is utilized to discuss security more exactly (exact security).

Definition 6. [Breaking Algorithm] *Let $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. We say that an adversary A is a $(t, q_H, q_{\mathcal{D}}, \epsilon)$-breaker for $\Pi(1^k)$ in GOAL-ATK if $Adv^{goal\text{-}atk}_{A,\Pi}(k) \geq \epsilon$ and, moreover, A runs within at most running time t, asking at most q_H queries to $H(\cdot)$ and at most $q_{\mathcal{D}}$ queries to $\mathcal{D}_{sk}(\cdot)$. In addition, q_H denotes the number of queries A asks to random function $H(\cdot)$, and similarly, $q_{\mathcal{D}}$ denotes the number of queries A asks to decryption oracle $\mathcal{D}_{sk}(\cdot)$. In the case of $atk = cpa$, then $q_{\mathcal{D}} = 0$. In the case of the standard model, then $q_H = 0$.*

In the following, we will recall the notion of Plaintext Awareness and the main results.

Definition 7. [Plaintext Awareness (PA)] *Let $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme, let B be an adversary, and let K be an polynomial-time algorithm (say knowledge extractor) . For any $k \in \mathbb{N}$ let*

$$Succ^{pa}_{K,B,\Pi}(k) := \Pr[H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}(1^k); \\ (\tau, \eta, y) \leftarrow runB^{H, \mathcal{E}_{pk}}(pk) : K(\tau, \eta, y, pk) = \mathcal{D}_{sk}(y)],$$

where $\tau := \{(h_1, H_1), \ldots, (h_{q_H}, H_{q_H})\}$, $\eta := \{y_1, \ldots, y_{q_E}\}$, and $y \notin \eta$. We describe a supplementary explanation: By $(\tau, \eta, y) \leftarrow runB^{H, \mathcal{E}_{pk}}(pk)$ we mean the following. Run B on input pk and oracles $H(\cdot)$ and $\mathcal{E}_{pk}(\cdot)$ and record (τ, η, y)

from B's interaction with its oracles. τ denotes the set of all B's queries and the corresponding answers of $H(\cdot)$. η denotes the set of all the answers (ciphertexts) received as the result of $\mathcal{E}_{pk}$. Here we insist that η doesn't include the corresponding queries (plaintexts) from B. y denotes the output of B.

We say that K is a $(t, \lambda(k))$-knowledge extractor if $Succ^{pa}_{K,B,\Pi}(k) \geq \lambda(k)$ and K runs within at most running time t (or t steps).

We say that Π is secure in the sense of PA if Π is secure in the sense of IND-CPA and there exists a $(t, \lambda(k))$-knowledge extractor K where t is polynomial in k and $(1 - \lambda(k))$ is negligible in k.

The following results proven in [3] is important.

Proposition 3. *PA $\rightarrow$ IND-CCA2 in the random oracle model.*

Corollary 2. *PA $\rightarrow$ NM-CCA2 in the random oracle model.*

3 Basic Scheme

Suppose a public-key encryption scheme, $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$, exists which is semantically secure against every chosen-plaintext (passive) attack (IND-CPA). Let $k_0(\cdot)$, $l_0(\cdot)$, $l(\cdot)$ and $n(\cdot)$ be positive integer valued functions bounded in some polynomial, namely $k_0(k), l_0(k), l(k), n(k) < \exists poly(k)$ for enough large k. We denote by $\Pi(1^{k+k_0}) = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ a public-key encryption scheme with $(k+k_0(k))$-bit length plaintext space, $l(k+k_0(k))$-bit length random value space and $n(k + k_0(k))$-bit length ciphertext space:

$$\mathcal{E}_{pk} : \{0,1\}^{k+k_0} \times \{0,1\}^{l} \rightarrow \{0,1\}^{n} \text{ and } \mathcal{D}_{sk} : \{0,1\}^{n} \rightarrow \{0,1\}^{k+k0},$$

where we write k_0, l, and n for $k_0(k)$, $l(k + k_0(k))$ and $n(k + k_0(k))$. In public-key encryption scheme Π, the (encryption) map from the plaintext space to the ciphertext space is one-to-one (injective). In addition, we define by

$$l_0(k + k_0) := \log_2(\min_{x \in \{0,1\}^{k+k_0}} [\#\{\mathcal{E}_{pk}(x, r) | r \in \{0,1\}^{l}\}])$$

the minimum number of the cardinality of encrypted values for fixed plaintext x. We often write l_0 for $l_0(k + k_0)$ for simplicity. Furthermore, we define by $H : \{0,1\}^{k+k_0} \rightarrow \{0,1\}^{l(k+k_0)}$ an ideal hash function.

We introduce a new public-key encryption scheme, $\bar{\Pi} := (\bar{\mathcal{K}}, \bar{\mathcal{E}}, \bar{\mathcal{D}})$ which is derived from Π and hash function H as follows:

Basic Scheme $\bar{\Pi} := (\bar{\mathcal{K}}, \bar{\mathcal{E}}, \bar{\mathcal{D}})$

- $\bar{\mathcal{K}}(1^k) := \mathcal{K}(1^{k+k_0})$ where k_0 denotes $k_0(k)$ for simplicity.
- $\bar{\mathcal{E}}_{pk} : \{0,1\}^{k} \times \{0,1\}^{k_0} \rightarrow \{0,1\}^{n}$ is defined by

$$\bar{\mathcal{E}}_{pk}(x, r) := \mathcal{E}_{pk}(x||r, H(x||r)),$$

where $|x| = k$, $|r| = k_0$, and $n := n(k + k_0(k))$.

– $\bar{\mathcal{D}}_{sk}(y) : \{0,1\}^n \to \{0,1\}^k$ is defined by

$$\bar{\mathcal{D}}_{sk}(y) := \begin{cases} [\mathcal{D}_{sk}(y)]^k & \text{if } y = \mathcal{E}_{pk}(\mathcal{D}_{sk}(y), H(\mathcal{D}_{sk}(y))) \\ \varepsilon \text{ (null)} & \text{otherwise} \end{cases}$$

where $[\mathcal{D}_{sk}(y)]^k$ denotes the first k-bit of $\mathcal{D}_{sk}(y)$.

Hereafter we will show that $\bar{\Pi}$ is semantically secure against every adaptive chosen-ciphertext attack.

4 Security

In this section, our goal is to prove Theorem 3. This theorem doesn't only show that if Π is IND-CPA secure then $\bar{\Pi}$ is IND-CCA2, but also show the exact reduction cost from $\bar{\Pi}$ to Π. The proof of Theorem 3 is derived from Theorems, 1 and 2.

We begin by showing Theorem 1. Recall that $k_0(\cdot)$, $l_0(\cdot)$, $l(\cdot)$ and $n(\cdot)$ are functions bounded in some polynomial, namely $k_0(k), l(k), n(k) < \exists poly(k)$ for enough large k and, for simplicity, we often use k_0, l_0, l, and n for $k_0(k)$, $l_0(k + k_0(k))$, $l(k + k_0(k))$ and $n(k + k_0(k))$.

Theorem 1. [Knowledge extractor K of $\bar{\Pi}$] *If there exists a (t, q_H)-adversary B, then there exist a constant c_0 and a $(t', \lambda(k))$-knowledge extractor K such that*

$$t' = t + q_H(T_{\mathcal{E}}(k) + c_0 \cdot k) \text{ and}$$
$$\lambda(k) = 1 - 2^{-l_0}.$$

Here $T_{\mathcal{E}}(k)$ denotes the computational running time of the encryption algorithm $\mathcal{E}_{pk}(\cdot)$ and $l_0 := \log_2(\min_{x \in \{0,1\}^{k+k_0}}[\#\{\mathcal{E}_{pk}(x, r) | r \in \{0,1\}^l\}])$.

Proof. The specification of knowledge extractor K is as follows:

Extractor: $K(\tau, \eta, y, pk)$
for q_H times do
if $y == \mathcal{E}_{pk}(h_i, H_i)$;
then $x \leftarrow [h_i]^k$ and break
else $x \leftarrow \varepsilon$ (null)
return x

End.

Here note that $\tau := \{(h_1, H_1), \ldots, (h_{q_H}, H_{q_H})\}$.

Now we define c_0 as corresponding to the computation time of comparing a bit to a bit plus some overhead, which depends on details of the underlying model of computation of K. Then, from the specification, K runs within $t + q_H(T_{\mathcal{E}}(k) + c_0 \cdot k)$ time.

Next we think of the probability that K outputs the plaintext, x, correctly, namely $x = \bar{\mathcal{D}}_{sk}(y)$. Here let $Fail$ be an event assigned to be true iff $x \neq \bar{\mathcal{D}}_{sk}(y)$

and let $AskH$ be an event assigned to be true iff there exists (h_i, H_i) in the list τ such that $y = \mathcal{E}_{pk}(h_i, H_i)$. Then it follows that

$$\begin{aligned}\Pr[Fail] &= \Pr[Fail|AskH] \cdot \Pr[AskH] + \Pr[Fail|\neg AskH] \cdot \Pr[\neg AskH] \\ &\leq \Pr[Fail|AskH] + \Pr[Fail|\neg AskH] \leq 0 + 2^{-l_0} = 2^{-l_0}.\end{aligned}$$

If $AskH$ is true then K never fail to guess the plaintext x and hence it is clear that $\Pr[Fail|AskH] = 0$.

Next in the case that $\neg AskH$ is true, K outputs ε: K guess y as *invalid*. Therefore, the probability of K's failure is that of B outputting *valid* y. We explain that $\Pr[Fail|\neg AskH]$ is at most 2^{-l_0} in the following.

Let us define event *good* y by being true iff $\mathcal{D}_{sk}(y) \neq \varepsilon$. Don't confuse it with *valid* y: *valid* y is defined to be true iff $\bar{\mathcal{D}}_{sk}(y) \neq \varepsilon$. Then note that

$$\begin{aligned}\Pr[P] := \Pr[Fail|\neg AskH] &= \Pr[P|\text{ good } y] \cdot \Pr[\textit{good } y] \\ &+ \Pr[P|\neg \textit{good } y] \cdot \Pr[\neg \textit{good } y] \leq \Pr[P|\textit{good } y].\end{aligned}$$

Therefore, it is enough to think of $\Pr[P|\ \textit{good}\ \ y]$.

Recall that $l_0 := \log_2(\min_{x\in\{0,1\}^{k+k_0}}[\#\{\mathcal{E}_{pk}(x,r)|r \in \{0,1\}^l\}])$. For *good* y, let define by $\mathcal{H}_y$ the set of $(h, \hat{H}_j)$'s such that $y = \mathcal{E}_{pk}(h, \hat{H}_j)$. Here $j \in \{1, \ldots, s\}$ and $s \leq 2^{l-l_0}$. Then since $\eta := \{y_1, \ldots, y_{q_E}\}$ and $y \notin \eta$, it follows that $h \neq \mathcal{D}_{sk}(y_i)$ for every $y_i \in \eta$. Therefore, for fixed *good* y (and h), since B doesn't ask query h to oracle $H(\cdot)$,

$$\Pr[P|\ \textit{good}\ \ y] = \Pr_{H\leftarrow\Omega}[H(h) \in \mathcal{H}_y] = s \cdot 2^{-l} \leq 2^{-l_0}.$$

This means that

$$\Pr[P] := \Pr[Fail|\neg AskH] \leq \Pr[P|\textit{good } y] \leq 2^{-l_0}.$$

Hence, $\lambda(k) = 1 - \Pr[Fail] = 1 - 2^{-l_0}$.

Theorem 2. **[$\bar{\Pi}$: IND-CPA secure]** *If there exists a $(t, q_H, 0, \epsilon)$-breaker $A := (A_1, A_2)$ for $\bar{\Pi}(1^k)$ in the sense of IND-CPA in the RO model, then there exist a constant c_1 and a $(t', 0, 0, \epsilon')$-breaker $A' := (A'_1, A'_2)$ for $\Pi(1^{k+k_0})$ in the sense of IND-CPA (in the standard model) where*

$$t' = t + c_1 \cdot q_H \cdot k, \ \textit{and}\ \epsilon' = \epsilon - q_H \cdot 2^{-(k_0-1)}.$$

Proof. We run $A' := (A'_1, A'_2)$ in the IND-CPA and standard model setting, using $A := (A_1, A_2)$ as oracles respectively.

Basically, when A_i asks query h, A'_i works as follows: If h has not been entered in list τ, A'_i, choosing l-bit random string H, makes an entry of (h, H) in τ and answers A_i with H. If (h, H) is already in list τ, A'_i answers A_i with the corresponding H. The list τ is empty at first. When A_1 outputs (x_0, x_1, s), A'_1 outputs $(x_0||r_0, x_1||r_1, s)$ where r_0, r_1 are k_0-bit random strings generated by A'_1.

Then, outside A', $y := \mathcal{E}_{pk}(X_b, R)$ is computed using a random bit $b \in \{0,1\}$ and l-bit random string R, where $X_0 := (x_0||r_0)$ and $X_1 := (x_1||r_1)$. y is inputted on A'_2 as well as (X_0, X_1, s).

If A_2 asks either X_0 or X_1 as a query, A'_2 makes A_2 stop and outputs the corresponding $b \in \{0,1\}$ as an answer, otherwise A_2 follows the basic rule mentioned above. When A_2 asks neither of them, A'_2 outputs b that A_2 output as an answer.

The argument behind the proof is as follows: If A_2 asks a query to A'_2, which coincides with either $(x_0||r_0)$ or $(x_1||r_1)$, it is almost equivalent to $\mathcal{D}_{sk}(y)$, because (even unbounded powerful) A_2 has no clue to k_0-bit random string $r_{\bar{b}}$, where $\bar{b}$ is the complement of bit b. Therefore, if A_2 asks either of them, the corresponding b is expected to be *valid*. On the other hand, if A_2 asks neither of them, A_2 is expected to output *valid* b because A_2 cannot distinguish y from a correct ciphertext for A_2.

The specification of adversary $A' := (A'_1, A'_2)$ is as follows:

Adversary: $A'_1(pk)$
$\tau \leftarrow \varepsilon$;
run $A_1(pk)$
do while A_1 does not make H-query h.
if $h \notin \tau_h$, where τ_h is the list of h's in τ
$H \leftarrow_R \{0,1\}^l$;
put (h, H) on the list τ;
answer A_1 with H;
else $h \in \tau_h$
answer A_1 with H such that $(h, H) \in \tau$
A_1 outputs (x_0, x_1, s)
$r_0, r_1 \leftarrow_R \{0,1\}^{k_0}$;
return $(x_0||r_0, x_1||r_1, s)$
End.

Adversary: $A'_2(x_0||r_0, x_1||r_1, s, y)$
run $A_2(x_0, x_1, s, y)$
do while A_1 does not make H-query h.
if $h == (x_b||r_b)$ for $b \in \{0,1\}$
stop A_2 and output b
else if $h \notin \tau_h$, where τ_h is the list of h's in τ
$H \leftarrow_R \{0,1\}^l$;
put (h, H) on the list τ;
answer A_1 with H;
else $h \in \tau_h$
answer A_1 with H such that $(h, H) \in \tau$
A_2 outputs b
return b
End.

Here, from Definition 4, b is chosen from $\{0,1\}$ with probability $1/2$, R is an l-bit random string, and $y = \mathcal{E}_{pk}(x_b||r_b, R)$.

c_1 corresponds to the computational time of comparing a bit to a bit, coin-flipping, and some overhead, depending on details of the underlying model of computation of A'. Then, from the specification of A', it runs within at most running time $(t + c_1 \cdot q_H \cdot k)$.

We now analyze the success probability of adversary $A' := (A'_1, A'_2)$. First we define the following events:

$$\begin{aligned}
SuccA := [&H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}(1^{k+k_0}); (x_0, x_1, s) \leftarrow A_1^H(pk); b \leftarrow_R \{0,1\};\\
&r_b, r_{\bar{b}} \leftarrow_R \{0,1\}^{k_0}; y \leftarrow \mathcal{E}_{pk}((x_b||r_b), H(x_b||r_b)) : A_2^H(x_0, x_1, s, y) = b],\\
SuccA' := [&(pk, sk) \leftarrow \mathcal{K}(1^{k+k_0}); (X_0, X_1, s) \leftarrow A'_1(pk'); b \leftarrow_R \{0,1\};\\
&R_b, R_{\bar{b}} \leftarrow_R \{0,1\}^{k_0}; y \leftarrow \mathcal{E}_{pk}(X_b, R_b) : A'_2(X_0, X_1, s, y) = b],
\end{aligned}$$

where $\bar{b}$ denotes the complement of b.

We can define the advantages of A and A', without loss of generality, as $Adv^{\text{ind-atk}}_{A,\Pi}(k+k_0) := 2 \cdot \Pr[SuccA] - 1$, and $Adv^{\text{ind-atk}}_{A',\Pi}(k) := 2 \cdot \Pr[SuccA'] - 1$.

Next, let us define by $Ask0$ an event assigned to be true iff a query of A_2 coincides with $(x_b||r_b)$ and by $Ask1$ an event assigned to be true iff a query of A_2 coincides with $(x_{\bar{b}}||r_{\bar{b}})$. Then,

$$\begin{aligned}
\Pr[SuccA] = &\Pr[SuccA|Ask0] \cdot \Pr[Ask0] + \Pr[SuccA|(\neg Ask0) \wedge Ask1]\\
&\cdot \Pr[(\neg Ask0) \wedge Ask1] + \Pr[SuccA|(\neg Ask0) \wedge (\neg Ask1)]\\
&\cdot \Pr[(\neg Ask0) \wedge (\neg Ask1)] \text{ and}\\
\Pr[SuccA'] = &\Pr[SuccA'|Ask0] \cdot \Pr[Ask0] + \Pr[SuccA'|(\neg Ask0) \wedge Ask1]\\
&\cdot \Pr[(\neg Ask0) \wedge Ask1] + \Pr[SuccA'|(\neg Ask0) \wedge (\neg Ask1)]\\
&\cdot \Pr[(\neg Ask0) \wedge (\neg Ask1)].
\end{aligned}$$

From the specification of A', it is clear that $\Pr[SuccA'|Ask0] = 1$, $\Pr[SuccA'|(\neg Ask0) \wedge Ask1] = 0$ and $\Pr[SuccA|(\neg Ask0) \wedge (\neg Ask1)] = \Pr[SuccA'|(\neg Ask0) \wedge (\neg Ask1)]$. Hence, $\Pr[SuccA']$ is at most $\Pr[(\neg Ask0) \wedge Ask1])$ less than $\Pr[SuccA]$ because

$$\begin{aligned}
\Pr[SuccA'] - \Pr[SuccA] &= (1 - \Pr[SuccA|Ask0]) \cdot \Pr[Ask0]\\
&\quad - \Pr[SuccA|(\neg Ask0) \wedge Ask1] \cdot \Pr[(\neg Ask0) \wedge Ask1]\\
&\geq - \Pr[(\neg Ask0) \wedge Ask1].
\end{aligned}$$

Finally, we have

$$\Pr[SuccA'] \geq \frac{\epsilon + 1}{2} - \frac{q_H}{2^{k_0}},$$

since we infer that $\Pr[(\neg Ask0) \wedge Ask1] \leq \frac{q_H}{2^{k_0}}$,

Therefore, we have that $\epsilon' = \epsilon - \frac{q_H}{2^{k_0 - 1}}$.

From Definition 7 and Theorems, 1 and 2, Π is secure in the sense of PA, and hence, by Proposition 3, secure in the sense of IND-CCA2. Thus, our interest in the following theorem is focused on the efficiency of the reduction.

Theorem 3. **[$\bar{\Pi}$: IND-CCA2 secure]** *If there exists a (t, q_H, q_D, ϵ)-breaker $A := (A_1, A_2)$ for $\bar{\Pi}(1^k)$ in the sense of IND-CCA2 in the RO model, then there exist constants, c, and $(t', 0, 0, \epsilon')$-breaker $A' := (A'_1, A'_2)$ for $\Pi(1^{k+k_0})$ in the sense of IND-CPA (in the standard model) where*

$$t' = t + q_H \cdot (T_{\mathcal{E}}(k) + c \cdot k), \text{ and}$$
$$\epsilon' = (\epsilon - q_H \cdot 2^{-(k_0-1)}) \cdot (1 - 2^{-l_0})^{q_D}.$$

$T_{\mathcal{E}}(k)$ denotes the computational running time of the encryption algorithm $\mathcal{E}_{pk}(\cdot)$ and $l_0 := \log_2(\min_{x\in\{0,1\}^{k+k_0}}[\#\{\mathcal{E}_{pk}(x, r)|r \in \{0,1\}^l\}])$.

c corresponds to $c_0 + c_1$. We omit the proof because it is straightforward from the following specification of adversary A':

Adversary: $A'_1(pk)$
$\tau \leftarrow \varepsilon$;
$\eta \leftarrow \varepsilon$;
run $A_1^{\mathcal{D}_{sk},H}(pk)$
  do while A_1 makes neither H-query h nor D-query y'
    if A_1 makes H-query h.
      if $h \notin \tau_h$
        $H \leftarrow_R \{0,1\}^l$;
        put (h, H) on the list τ;
        answer A_1 with H;
      else $h \in \tau_h$
        answer A_1 with H such that $(h, H) \in \tau$
    else if A_1 makes D-query y'.
      run $K(\tau, \eta, y', pk)$
        K outputs x'
      answer A_1 with x'
  A_1 outputs (x_0, x_1, s)
$r_0, r_1 \leftarrow_R \{0,1\}^{k_0}$;
return $(x_0||r_0, x_1||r_1, s)$
End.

Adversary: $A'_2(x_0||r_0, x_1||r_1, s, y)$
$\eta \leftarrow y$;
run $A_2^{\mathcal{D}_{sk},H}(x_0, x_1, s, y)$
  do while A_1 makes neither H-query h nor D-query y'
    if A_1 makes H-query h
      if $[h]_{k_0} == r_b$, where $[h]_{k_0}$ the last k_0-bit of h
        stop A_1 and output b
      else if $h \notin \tau_h$

$H \leftarrow_R \{0,1\}^l$;
put (h, H) on the list τ;
answer A_1 with H;
else $h \in \tau_h$
answer A_1 with H such that $(h, H) \in \tau$
else if A_1 makes D-query y'
run $K(\tau, \eta, y', pk)$
K outputs x'
answer A_1 with x'
A_1 outputs b
return b

End.

5 Examples: Enhanced Probabilistic Encryptions

In this section, we convert IND-CPA secure ones to IND-CCA2 (or NM-CCA2) secure ones. The ElGamal, Okamoto-Uchiyama, and Blum-Goldwasser encryption schemes [4,7,9] are candidates, since they are practical and secure in the IND-CPA sense under some reasonable assumptions; the decision Diffie-Hellman [2], p-subgroup, and factoring assumptions, respectively.

[Enhanced ElGamal scheme]

- Key-generator $\bar{\mathcal{K}}$: $(pk, sk) \leftarrow \bar{\mathcal{K}}(1^k) := \mathcal{K}(1^{k+k_0(k)})$
- $pk := (p, q, g, y)$ and $sk := (p, q, g, s)$ where $y = g^s \bmod p$, $|p| = k + k_0$, $s \in \mathbb{Z}/q\mathbb{Z}$, $q|p-1$, and $\# < g >= q$.
- Hash function H: $\{0,1\}^{k+k_0} \longrightarrow \mathbb{Z}/q\mathbb{Z}$.
- Encryption $\bar{\mathcal{E}}$:

$$(y_1, y_2) := \bar{\mathcal{E}}_{pk}(x, r) := (g^{H(x||r)} \bmod p, (x||r) \oplus (y^{H(x||r)} \bmod p)),$$

  where message $x \in \{0,1\}^k$ and $r \leftarrow_R \{0,1\}^{k_0}$.
- Decryption $\bar{\mathcal{D}}$:

$$\bar{\mathcal{D}}_{sk}(y_1, y_2) := \begin{cases} [y_2 \oplus (y_1^s \bmod p)]^k & \text{if } y_1 = g^{H(y_2 \oplus (y_1^s \bmod p))} \bmod p \\ \varepsilon \text{ (null)} & \text{otherwise} \end{cases}$$

  where $[y_2 \oplus (y_1^s \bmod p)]^k$ denotes the first k-bit of $y_2 \oplus (y_1^s \bmod p)$.

Lemma 1. *In the random oracle model, the Enhanced ElGamal encryption scheme is secure in the sense of NM-CCA2 (or IND-CCA2) if the decision Diffie-Hellman problem is intractable.*

[2] To our knowledge, Tsiounis and Yung first proved in [11] that the ElGamal encryption scheme is as secure as the decision Diffie-Hellman problem. In addition, they also presented a converted ElGamal scheme which is NM-CCA2 secure in the random oracle model. However, our converted one is more efficient than theirs.

[Enhanced Okamoto-Uchiyama scheme]

- Key-generator $\bar{\mathcal{K}}$: $(pk, sk) \leftarrow \bar{\mathcal{K}}(1^k) := \mathcal{K}(1^{k+k_0(k)})$
- $pk := (n, g, h, k)$ and $sk := (p, q)$ where $n = p^2q$, $|p| = |q| = k + k_0$, $g \in (\mathbb{Z}/n\mathbb{Z})^*$ such that the order of $g_p := g^{p-1} \bmod p^2$ is p, and $h = g^n \bmod n$.
- Hash function H: $\{0,1\}^{k+k_0-1} \longrightarrow \mathbb{Z}/n\mathbb{Z}$.
- Encryption $\bar{\mathcal{E}}$:

$$y := \bar{\mathcal{E}}_{pk}(x, r) := g^{(x||r)} h^{H(x||r)} \bmod n,$$

  where message $x \in \{0,1\}^k$ and $r \leftarrow_R \{0,1\}^{k_0-1}$.
- Decryption $\mathcal{D}$:

$$\bar{\mathcal{D}}_{sk}(y) := \begin{cases} [\frac{L(y_p)}{L(g_p)} \bmod p]^k & \text{if } y = g^X h^{H(X)} \bmod n \\ \varepsilon \text{ (null)} & \text{otherwise} \end{cases}$$

  where $y_p := y^{p-1} \bmod p^2$, $L(x) := \frac{x-1}{p}$, and $X := \frac{L(y_p)}{L(g_p)} \bmod p$.

Lemma 2. *In the random oracle model, the Enhanced Okamoto-Uchiyama encryption scheme is secure in the sense of NM-CCA2 (or IND-CCA2) if the p-subgroup problem (see [9]) is intractable.*

[Enhanced Blum-Goldwasser scheme]

- Key-generator $\bar{\mathcal{K}}$: $(pk, sk) \leftarrow \bar{\mathcal{K}}(1^k) := \mathcal{K}(1^{k+k_0(k)})$
- $pk := (n)$ and $sk := (n, p, q)$ where $n = pq$, $|p| = |q| = k/2$, and p, q are William integers (i.e. $p, q \equiv 7 \pmod 8$ and primes).
- Hash function H: $\{0,1\}^{k+k_0} \longrightarrow \mathbb{Z}/n\mathbb{Z}$.
- Encryption $\bar{\mathcal{E}}$:

$$(y_1, y_2) := \bar{\mathcal{E}}_{pk}(x, r) := (H(x||r)^{2^{k+1}} \bmod n,\ x \oplus R).$$

  where message $x \in \{0,1\}^k$, $r \leftarrow_R \{0,1\}^{k_0}$, and $R := LSB[H(x||r)^2] \,||LSB[H(x||r)^{2^2}] \,||\cdots||\, LSB[H(x||r)^{2^k}]$.
- Decryption $\bar{\mathcal{D}}$:

$$\bar{\mathcal{D}}_{sk}(y_1, y_2) := \begin{cases} [y_2 \oplus \hat{R}]^k & \text{if } y_1 = H(y_2 \oplus \bar{R})^{2^{k+1}} \bmod n \\ \varepsilon \text{ (null)} & \text{otherwise} \end{cases}$$

  where $\hat{R} := LSB[y_1^{2^{-k}}]||\cdots||LSB[y_1^{2^{-1}}]$.

Lemma 3. *In the random oracle model, the Enhanced Blum-Goldwasser encryption scheme is secure in the sense of NM-CCA2 (or IND-CCA2) if the factoring problem is intractable.*

6 Conclusion

This paper presented a simple and efficient conversion from a semantically secure public-key encryption scheme against *passive adversaries* to a non-malleable (or semantically secure) public-key encryption scheme against *chosen-ciphertext attacks* (*active adversaries*) in the random oracle model. Our conversion incurs minimum cost, i.e., only one random (hash) function operation. We also showed that our security reduction is (almost) optimally efficient, or exact security. Finally this paper presented some practical examples, the enhanced ElGamal, Blum-Goldwasser and Okamoto-Uchiyama schemes.

Acknowledgment

The second author would like to thank Phillip Rogaway for useful discussions.

References

1. M. Bellare and P. Rogaway, "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols," Proc. of the First ACM Conference on Computer and Communications Security, pp.62–73.
2. M. Bellare and P. Rogaway, "Optimal Asymmetric Encryption—How to encrypt with RSA" Advances in Cryptology –EUROCRYPT'94.
3. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations Among Notions of Security for Public-Key Encryption Schemes" Advances in Cryptology –CRYPTO'98.
4. M. Blum, and S. Goldwasser, "An efficient probabilistic public-key encryption scheme which hides all partial information", Proceeding of Crypto'84, LNCS 196, Springer-Verlag, pp.289-299 (1985).
5. R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen message attack", Advances in Cryptology –CRYPTO'98, Springer-Verlag, 1998.
6. D. Dolev and C. Dwork and M. Naor, "Non-malleable cryptography", Proceeding of STOC91, pp 542–552.
7. T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Transactions on Information Theory, IT-31, 4, pp.469–472, 1985.
8. S. Goldwasser, and S. Micali, "Probabilistic Encryption", JCSS, vol.28, pp.270–299, 1984.
9. T. Okamoto, and S. Uchiyama, "A New Public-Key Cryptosystem as Secure as Factoring", Advances in Cryptology –EUROCRYPT'98, Springer-Verlag, 1998.
10. R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", Communications of ACM, 21, 2, pp.120-126, 1978.
11. Y. Tsiounis and M. Yung, "On the Security of ElGamal based Encryption", PKC'98, January, 1998.