Secure Integration of Asymmetric and Symmetric Encryption Schemes

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 shows a generic and simple conversion from weak asymmetric and symmetric encryption schemes into an asymmetric encryption scheme which is secure in a very strong sense — indistinguishability against adaptive chosen-ciphertext attacks in the random oracle model. In particular, this conversion can be applied efficiently to an asymmetric encryption scheme that provides a large enough coin space and, for every message, many enough variants of the encryption, like the ElGamal encryption scheme.

Key words: Indistinguishability, Adaptive chosen-ciphertext attack, Random oracle model, Hybrid encryption.

1 Introduction

Suppose that an asymmetric encryption scheme is secure in a very weak sense — an adversary can't entirely decrypt the encryption of a random plaintext. Suppose that a symmetric encryption scheme is secure in the following weak sense — for all possible messages, m_1 and m_2, in the indicated message space, an adversary can't distinguish the encryption of m_1 from the encryption of m_2 (where the adversary is not given the ability to encrypt or decrypt desired strings). From these schemes, we construct a new asymmetric encryption scheme. The (hybrid) encryption of a plaintext m is

$$\mathcal{E}^{\mathrm{hy}}_{pk}(m) = \quad \mathcal{E}^{\mathrm{asym}}_{pk}(\sigma; H(\sigma, m)) \quad || \quad \mathcal{E}^{\mathrm{sym}}_{G(\sigma)}(m),$$

where

- σ is a random string chosen from an appropriate domain,
- $\mathcal{E}^{\mathrm{asym}}_{pk}(\text{message}; \text{coins})$ indicates the asymmetric encryption of the indicated message using the indicated coins as random bits,
- $\mathcal{E}^{\mathrm{sym}}_{a}(\text{messgage})$ indicates the symmetric encryption of the indicated message using the indicated string a, and
- G and H denote hash functions.

Michael Wiener (Ed.): CRYPTO'99, LNCS 1666, pp. 537-554, 1999.

In the random oracle model (namely, G and H are modeled as random oracles), this hybrid encryption scheme is secure in a very strong sense — indistinguishability against adaptive chosen-ciphertext attacks.

We will provide the concrete security reduction in the exact security manner [3]. The security of this hybrid encryption scheme depends only on those of asymmetric and symmetric encryption primitives and the following property of the asymmetric encryption primitive — given an appropriate message space, for any message in the space, the variants of the encryption occur in a large *enough* number, provided the coins are chosen uniformly from the coin space of the encryption scheme. We will define the exact definition later and will also show, for any encryption scheme, a slightly modified scheme can provide an enough number of the variants. In particular, this conversion can be efficiently applied to an asymmetric encryption scheme with a large coin space, like the ElGamal encryption scheme.

1.1 Related Works

To create a practical and provably secure encryption scheme is one of important goals in cryptography. Although theoretical works have been done in many literatures [18, 13, 14, 19, 10], there are not so many schemes that satisfy both provable security and efficiency. In this section, we will refer to several schemes that are practical and provably secure in a very strong sense, such as [23, 3, 8, 1, 12], and will discuss these schemes (including ours) in the following terms.

Conversion A promising way to construct a practical and provably-secure encryption scheme is to convert it from primitives which are secure in a weaker sense.

In CRYPTO'94, Bellare and Rogaway presented a generic and simple conversion from a one-way trapdoor permutation (OWTP) such as the RSA primitive into an asymmetric encryption scheme which is secure in a very strong sense in the random oracle model [3]. A scheme created in this way is called OAEP (Optimal Asymmetric Encryption Padding). The strong security notion is *indistinguishability against adaptive chosen-ciphertext attacks* (IND-CCA), as described in [19]. However, the method in [3] was not applied to asymmetric encryption schemes. Therefore, several (practical) asymmetric encryption schemes lie outside the range of OAEP conversion, e.g., the ElGamal, Blum-Goldwasser, and Okamoto-Uchiyama encryption schemes [11, 6, 17].

Before their proposal, Zheng and Seberry had also proposed some practical schemes [23] aiming at chosen-cipher security, and, according to [3], in the random oracle model at least one of their schemes enjoys the same security as OAEP. That scheme too is, however, applied only to OWTPs.

The current authors recently presented a generic conversion from an asymmetric encryption scheme into an asymmetric one that is secure in the IND-CCA sense in the random oracle model [12]. However the security requirement of the primitive encryption scheme is stronger than that of [3] — the OAEP conversion starts from a OWTP while the conversion in [12] does from an asymmetric

encryption scheme which is secure in the sense of indistinguishability against chosen-plaintext attacks (IND-CPA).

Other conversion have been reported, such as [1, 22, 21]. However their schemes depend strongly on the primitive encryptions, they don't work for generic methods.

To the best of our knowledge, there has up to now not been proposed any generic (and efficient) method to convert an asymmetric encryption scheme into an IND-CCA secure one.

Hybrid Encryption An asymmetric encryption scheme is usually employed only for distributing a secret-key of a symmetric encryption scheme for message encryption. Actually, the hybrid usage of asymmetric and symmetric encryption schemes is very common in practice. On the other hand, hybrid usage is insecure in general, even if both the asymmetric and symmetric encryption schemes are secure in very strong senses. In spite of the fact that hybrid usage is common and that, in general, this is insecure, there has been little research on this subject; see [1, 8].

In [1], Abdalla, Bellare, and Rogaway present a hybrid encryption scheme, called DHAES, and prove that hybrid usage is secure in the IND-CCA sense in the random oracle model (or a strong assumption in the standard (not random oracle) model). The main difference from our work is that they use one more cryptographic primitive — message authentication code (MAC). In addition, their scheme depends on the Diffie-Hellman key-distribution scheme. DHAES is composed of the Diffie-Hellman key-distribution, a hash function, a symmetric encryption, and a message authentication code (MAC).

Cramer and Shoup briefly mentioned in their work that their scheme can be applied to hybrid usage with a symmetric encryption scheme [8].

1.2 Our Results

The contributions of this paper are twofold: One is to show a *generic* conversion from a very weak asymmetric encryption to an asymmetric encryption scheme which is secure in a very strong sense (IND-CCA in the random oracle model). The other is to exhibit a *generic* hybrid conversion of asymmetric and symmetric encryption schemes, proving the security explicitly. Our conversion starts from arbitrary encryption schemes and each scheme so obtained is approximately as efficient as, or more efficient than, the previously proposed schemes [1, 8, 22, 21].

2 Preliminary

We begin with some notations.

Definition 1. *Let A be a probabilistic algorithm and let $A(x_1, \ldots, x_n; r)$ be the result of running A on input $(x_1, \ldots, x_n)$ and random coins r. We denote by $y \leftarrow A(x_1, \ldots, x_n)$ the experiment of picking r at random and letting y be*

$A(x_1, \ldots, x_n; r)$ (i.e., $y = A(x_1, \ldots, x_n; r)$). If $\mathcal{S}$ is a finite set, let $y \leftarrow_R \mathcal{S}$ be the operation of picking y at random and uniformly from $\mathcal{S}$. When $S, T, \ldots$, denote probability spaces, $\Pr[x \leftarrow S; y \leftarrow T; \cdots : p(x, y, \ldots)]$ denotes the probability that the predicate, $p(x, y, \ldots)$, is true after the experiments, $x \leftarrow S, y \leftarrow T, \cdots$, are executed in that order. Moreover, $|x|$ denotes the bit length of string x and $\#S$ denotes the cardinality of set S.

Here we define asymmetric and symmetric encryption schemes, basically following [13, 3].

Definition 2. [**Asymmetric Encryption**] *An asymmetric encryption scheme, $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D}, \mathtt{COINS}, \mathtt{MSPC})$, is a triple of algorithm, associated with finite sets, $\mathtt{COINS}(k)$ and $\mathtt{MSPC}(k)$, $\subseteq \{0,1\}^*$, for $k \in \mathbb{N}$, where*

- *$\mathcal{K}$, called the key-generation algorithm, is a probabilistic algorithm which on input 1^k ($k \in \mathbb{N}$) outputs a pair of strings, $(pk, sk) \leftarrow \mathcal{K}(1^k)$.*
- *$\mathcal{E}$, called the encryption algorithm, is a probabilistic algorithm that takes a pair of strings, pk and x, and a string $r \leftarrow \mathtt{COINS}(k)$, and produces a string $y = \mathcal{E}_{pk}(x; r)$.*
- *$\mathcal{D}$, called the decryption algorithm, is a deterministic algorithm that takes a pair of strings, sk and y, and returns a string $x \leftarrow \mathcal{D}_{sk}(y)$.*

We require that, for any $k \in \mathbb{N}$, if $(pk, sk) \leftarrow \mathcal{K}(1^k)$, $x \in \mathtt{MSPC}$, and $y \leftarrow \mathcal{E}_{pk}(x)$, then $\mathcal{D}_{sk}(y) = x$.

Definition 3. [**Symmetric Encryption**] *A symmetric encryption scheme, $\Pi = (\mathcal{E}, \mathcal{D}, \mathtt{KSPC}, \mathtt{MSPC})$, is a pair of algorithms associated with finite sets, $\mathtt{KSPC}(k)$ and $\mathtt{MSPC}(k)$, $\subseteq \{0,1\}^*$, for $k \in \mathbb{N}$, where*

- *$\mathcal{E}$, called the encryption algorithm, is a deterministic algorithm that takes a pair of strings, a and x, and produces $y = \mathcal{E}_a(x)$.*
- *$\mathcal{D}$, called the decryption algorithm, is a deterministic algorithm that takes a pair of strings, a and y, and outputs a string $x = \mathcal{D}_a(y)$.*

We require that, for any $k \in \mathbb{N}$, if $a \in \mathtt{KSPC}(k)$, $x \in \mathtt{MSPC}$, and $y = \mathcal{E}_a(x)$, then $\mathcal{D}_a(y) = x$.

3 Basic Conversion

In this section, we present our conversion.

Let $\Pi^{\mathrm{asym}} = (\mathcal{K}^{\mathrm{asym}}, \mathcal{E}^{\mathrm{asym}}, \mathcal{D}^{\mathrm{asym}}, \mathtt{COINS}^{\mathrm{asym}}, \mathtt{MSPC}^{\mathrm{asym}})$ be an asymmetric encryption scheme and let $\Pi^{\mathrm{sym}} = (\mathcal{E}^{\mathrm{sym}}, \mathcal{D}^{\mathrm{sym}}, \mathtt{KSPC}^{\mathrm{sym}}, \mathtt{MSPC}^{\mathrm{sym}})$ be a symmetric encryption scheme. Let $G : \mathtt{MSPC}^{\mathrm{asym}} \to \mathtt{KSPC}^{\mathrm{sym}}$ and $H : \mathtt{MSPC}^{\mathrm{asym}} \times \mathtt{MSPC}^{\mathrm{sym}} \to \mathtt{COINS}^{\mathrm{asym}}$ be hash functions.

From these primitives, we present a new asymmetric encryption scheme, $\Pi^{\mathrm{hy}} = (\mathcal{K}^{\mathrm{hy}}, \mathcal{E}^{\mathrm{hy}}, \mathcal{D}^{\mathrm{hy}}, \mathtt{COINS}^{\mathrm{hy}}, \mathtt{MSPC}^{\mathrm{hy}})$, (where $\mathtt{COIN}^{\mathrm{hy}} = \mathtt{MSPC}^{\mathrm{asym}}$ and $\mathtt{MSPC}^{\mathrm{hy}} = \mathtt{MSPC}^{\mathrm{sym}}$) as follows:

– Encryption

$$\mathcal{E}^{\mathrm{hy}}_{pk}(m;\sigma) = \mathcal{E}^{\mathrm{asym}}_{pk}(\sigma; H(\sigma,m)) \;\;||\;\; \mathcal{E}^{\mathrm{sym}}_{G(\sigma)}(m).$$

– Decryption

$$\mathcal{D}^{\mathrm{hy}}_{sk}(c_1||c_2) = \begin{cases} \mathcal{D}^{\mathrm{sym}}_{G(\hat{\sigma})}(c_2) & \text{if } c_1 = \mathcal{E}^{\mathrm{asym}}_{pk}(\hat{\sigma}; H(\hat{\sigma},\hat{m})), \\ \perp & \text{otherwise.} \end{cases}$$

where $\hat{\sigma} := \mathcal{D}^{\mathrm{asym}}_{sk}(c_1)$ and $\hat{m} := \mathcal{D}^{\mathrm{sym}}_{G(\hat{\sigma})}(c_2)$. (If there isn't $\mathcal{D}^{\mathrm{asym}}_{sk}(c_1)$ or $\mathcal{D}^{\mathrm{sym}}_{G(\hat{\sigma})}(c_2)$, then $\mathcal{D}^{\mathrm{hy}}_{sk}(c_1||c_2) = \perp$.)

Key-Generation $\mathcal{K}^{\mathrm{hy}}(1^k)$	Encryption $\mathcal{E}^{\mathrm{hy}}_{pk}(m)$	Decryption $\mathcal{D}^{\mathrm{hy}}_{sk}(c_1,c_2)$
$(pk,sk) \leftarrow \mathcal{K}^{\mathrm{asym}}(1^k)$.	$\sigma \leftarrow_R \mathtt{MSPC}^{\mathrm{asym}}$.	$\hat{\sigma} := \mathcal{D}^{\mathrm{asym}}_{sk}(c_1)$.
	$r_1 := H(\sigma,m)$.	$\hat{r_2} := G(\hat{\sigma})$.
	$r_2 := G(\sigma)$.	$\hat{m} := \mathcal{D}^{\mathrm{sym}}_{\hat{r_2}}(c_2)$.
	$c_1 := \mathcal{E}^{\mathrm{asym}}_{pk}(\sigma; r_1)$.	$\hat{r_1} := H(\hat{\sigma},\hat{m})$.
	$c_2 \leftarrow \mathcal{E}^{\mathrm{sym}}_{r_2}(m)$.	If $c_1 == \mathcal{E}^{\mathrm{asym}}_{pk}(\hat{\sigma}; \hat{r_1})$ then $m := \hat{m}$, else $m := \perp$.
return (pk,sk).	return (c_1,c_2).	return m.

Fig. 1. Hybrid encryption scheme

4 Security Definitions

4.1 Asymmetric Encryption

In this section we define security notions for asymmetric encryption.

One-way Encryption In the following, we give a very weak security notion (one-wayness) for an asymmetric encryption. Let $\Pi = (\mathcal{K},\mathcal{E},\mathcal{D},\mathtt{COINS},\mathtt{MSPC})$ be an asymmetric encryption. For Π, we consider an algorithm, A, called an adversary, that, taking a public-key, pk, outputted by $\mathcal{K}$, and an encryption, y, of a random plaintext in MSPC tries to decrypt y. The probability of A's success, denoted by the advantage of A, depends on A, Π, and the random choice of a plaintext from MSPC. A doesn't have any decryption oracle (while an encryption oracle doesn't matter because chosen-plaintext attacks are clearly unavoidable in an asymmetric encryption scheme).

Definition 4. [**OWE**] *Let $\Pi = (\mathcal{K},\mathcal{E},\mathcal{D},\mathtt{COINS},\mathtt{MSPC})$ be an asymmetric encryption scheme. Let A be an adversary. For $k \in \mathbb{N}$, define the advantage of A by* $\mathrm{Adv}^{\mathrm{owe}}_{A,\Pi,\mathtt{MSPC}}(k) =$

$$\Pr[(pk,sk) \leftarrow \mathcal{K}(1^k); x \leftarrow \mathtt{MSPC}(k); y \leftarrow \mathcal{E}_{pk}(x) : A(pk,y) = \mathcal{D}_{sk}(y)].$$

We say that adversary A (t,ϵ)-breaks Π in the sense of OWE *if A runs in at most time t and achieves* $\mathrm{Adv}^{\mathrm{owe}}_{A,\Pi}(k) \geq \epsilon$. *We say that Π is (t,ϵ)-secure in the sense of* OWE *if there is no adversary that (t,ϵ)-breaks Π in that sense.*

γ-uniformity We introduce a property of asymmetric encryption in the following definition.

Definition 5. [**γ-uniformity**] *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D}, \mathtt{COINS}, \mathtt{MSPC})$ be an asymmetric encryption scheme. For given $(pk, sk) \in K(1^k)$, $x \in \mathtt{MSPC}$ and $y \in \{0,1\}^*$, define*

$$\gamma(x,y) = \Pr[h \leftarrow_R \mathtt{COINS} : y = \mathcal{E}_{pk}(x;h)].$$

We say that Π is γ-uniform (for $k \in \mathbb{N}$), if, for any $(pk, sk) \in K(1^k)$, any $x \in \mathtt{MSPC}$ and any $y \in \{0,1\}^*$, $\gamma(x,y) \leq \gamma$.

Example 6. Let k be a security parameter. Define by $((y,g,p,q),x)$ a pair of public-key and secret-key where $y = g^x \bmod p$, $x \in \mathbb{Z}/q\mathbb{Z}$, $q = \# < g >$ and k $= |q|$. The ElGamal encryption scheme, associated with 1^k, is then 2^{-k}-uniform.

Strong Security Notions We recall a classical and stronger security notion for an asymmetric encryption, called *indistinguishability* (IND), following [13, 4].

In this security notion, we consider an adversary, A, that takes two stages, find and guess. In the find stage, A takes public-key pk and returns two distinct messages, x_0, x_1, and a string, s, to use in the next mode, and then, in the guess stage, takes the encryption of x_b, where $b \leftarrow_R \{0,1\}$, and the above information, and tries to guess b. The advantage of A is meant by how well she can guess the value b. If A has the decryption oracle, $\mathcal{D}_{sk}(\cdot)$, we say that this experiment is an adaptive chosen-ciphertext attack (CCA), while, if A doesn't have it, we call it a chosen-plaintext attack (CPA).

The random oracle version of this security notion is defined by allowing A to make access to a random oracle (or plural oracles), which depends on Π. We define by Ω the map family from an appropriate domain to an appropriate range. The domain and range depend on the underlying encryption scheme, Π. Even if we choose two random functions that have distinct domains and distinct ranges respectively, we just write the experiment, for convenience, as $G, H \leftarrow \Omega$, instead of preparing two map families.

In the following definition, we define simultaneously indistinguishability with regard to CCA and CPA in the random oracle model.

Definition 7. [**Indistinguishability**] *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D}, \mathtt{COINS}, \mathtt{MSPC})$ be an asymmetric encryption scheme and let A be an Adversary. For $k \in \mathbb{N}$, define the following two advantages:*

– $\mathrm{Adv}^{\mathrm{ind-cpa}}_{A,\Pi}(k) =$

$$2 \cdot \Pr[G, H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}(1^k); (x_0, x_1, s) \leftarrow A^{G,H}(\mathrm{find}, pk); \\ b \leftarrow_R \{0,1\}; y \leftarrow \mathcal{E}^{G,H}_{pk}(x_b) : A^{G,H}(\mathrm{guess}, s, y) = b] - 1$$

– $\mathrm{Adv}^{\mathrm{ind-cca}}_{A,\Pi}(k) =$

$$2 \cdot \Pr[G, H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}(1^k); (x_0, x_1, s) \leftarrow A^{G,H,\mathcal{D}_{sk}}(\mathrm{find}, pk); \\ b \leftarrow_R \{0,1\}; y \leftarrow \mathcal{E}^{G,H}_{pk}(x_b) : A^{G,H,\mathcal{D}_{sk}}(\mathrm{guess}, s, y) = b] - 1.$$

We require that, for (x_0, x_1) that A outputs, $x_0 \neq x_1$ and $x_0, x_1 \in \mathtt{MSPC}$.

We say that adversary A (t, q_g, q_h, ϵ)-breaks Π in the sense of IND-CPA *in the random oracle model if A runs in at most time t, asks at most q_g queries to $G(\cdot)$, asks at most q_h queries to $H(\cdot)$, and achieves $\mathrm{Adv}^{\mathrm{ind-cpa}}_{A,\Pi}(k) \geq \epsilon$.*

Similarly, we say that adversary A $(t, q_g, q_h, q_d, \epsilon)$-breaks Π in the sense of IND-CCA *in the random oracle model if A runs in at most time t, asks at most q_g queries to $G(\cdot)$, asks at most q_h queries to $H(\cdot)$, asks at most q_d queries to $\mathcal{D}_{sk}(\cdot)$, and achieves $\mathrm{Adv}^{\mathrm{ind-cca}}_{A,\Pi}(k) \geq \epsilon$.*

We say that Π is (t, q_g, q_h, ϵ)-secure (or $(t, q_g, q_h, q_d, \epsilon)$-secure) in the sense of IND-CPA (*or* IND-CCA) *if there is no adversary that (t, q_g, q_h, ϵ)-breaks (or $(t, q_g, q_h, q_d, \epsilon)$-breaks) Π in the corresponding sense.*

Knowledge Extractor The notion of knowledge extractor for an asymmetric encryption scheme is defined in [3, 4]. We recall the definition, following [4]. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D}, \mathtt{COINS}, \mathtt{MSPC})$ be an asymmetric encryption scheme. Let B and K be algorithms, called an adversary and a knowledge extractor, respectively. They work in the random oracle model as follows:

– Adversary B takes public-key pk and asks two kinds of queries, queries for random oracles, G and H, and queries for an encryption oracle, $\mathcal{E}^{G,H}_{pk}$, and, after taking the answers from those oracles, finally outputs a string y, where
  - $\mathcal{T}_G$ denotes the set of all pairs of B's queries and the corresponding answers from G,
  - $\mathcal{T}_H$ denotes the set of all pairs of B's queries and the corresponding answers from H,
  - $\mathcal{Y}$ denotes the set of all answers recieved as ciphertexts from $\mathcal{E}^{G,H}_{pk}(\cdot)$.
  - y (output of B) is not in $\mathcal{Y}$.

  We write the experiment above as $(\mathcal{T}_G, \mathcal{T}_H, \mathcal{Y}, y) \leftarrow B^{G,H,\mathcal{E}_{pk}}(pk)$.
  Here we insist that neither any query of B's to $\mathcal{E}_{pk}$ is in $\mathcal{Y}$, nor any query of $\mathcal{E}^{G,H}_{pk}$'s to random oracles, G and H, is in $\mathcal{T}_G$ and $\mathcal{T}_H$.
– Knowledge extractor K takes $(\mathcal{T}_G, \mathcal{T}_H, \mathcal{Y}, y, pk)$ and outputs a string x.

Definition 8. [Knowledge Extractor] *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D}, \mathtt{COINS}, \mathtt{MSPC})$ be an asymmetric encryption scheme, let B be an adversary, and let K be a knowledge extractor. Define the following advantage: For $k \in \mathbb{N}$, let* $\mathrm{Succ}^{\mathrm{ke}}_{K,B,\Pi}(k) =$

$$\Pr[G, H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}(1^k); (\mathcal{T}_G, \mathcal{T}_H, \mathcal{Y}, y) \leftarrow B^{G,H,\mathcal{E}_{pk}}(pk) : \\ K(\mathcal{T}_G, \mathcal{T}_H, \mathcal{Y}, y, pk) = \mathcal{D}_{sk}(y)].$$

We say that B is a (q_g, q_h, q_e)-adversary if B takes pk, makes at most q_g queries to G, at most q_h queries to H and at most q_e queries to $\mathcal{E}_{pk}$ respectively,

and finally produces a string y. We say that K is a (t,λ)-knowledge extractor for B if K takes $(\mathcal{T},\mathcal{Y},y)$, runs in at most time t, and achieves $\mathrm{Succ}^{\mathrm{ke}}_{K,B,\Pi}(k) \geq \lambda$.

4.2 Symmetric Encryption

We prepare a security notion for symmetric encryptions, called find-guess. This notion is the symmetric encryption version of indistinguishability, following [5].

Let $\Pi = (\mathcal{E},\mathcal{D},\mathtt{KSPC},\mathtt{MSPC})$ be a symmetric-key encryption scheme and let A be a probabilistic algorithm (called an adversary). In the find stage, adversary A outputs two distinct messages, x_0, x_1, and some information, s, to use in the next mode, then, in the guess stage, takes the encryption of x_b where $b \leftarrow_R \{0,1\}$ and the above information, and tries to guess b. The advantage of A is meant by how well she can guess the value b. In our definition, unlike [5], A doesn't have any encryption oracle.

Definition 9. [Find-Guess] *Let $\Pi = (\mathcal{E},\mathcal{D},\mathtt{KSPC},\mathtt{MSPC})$ be a symmetric-key encryption scheme and let A be an adversary. For $k \in \mathbb{N}$, define the advantage of A, by* $\mathrm{Adv}^{\mathrm{fg}}_{A,\Pi}(k) =$

$$2 \cdot \Pr[a \leftarrow_R \mathtt{KSPC}(k); (x_0,x_1,s) \leftarrow A(\mathrm{find}); b \leftarrow_R \{0,1\}; \\ y = \mathcal{E}_a(x_b) : A(\mathrm{guess},s,y) = b] - 1.$$

We require that, for (x_0,x_1) that A outputs, $x_0 \neq x_1$ and $x_0, x_1 \in \mathtt{MSPC}$.

We say that adversary A (t,ϵ)-breaks Π in the sense of FG *in the random oracle model if A runs in at most time t and achieves* $\mathrm{Adv}^{\mathrm{fg}}_{A,\Pi}(k) \geq \epsilon$. *We say Π is (t,ϵ)-secure in the sense of* FG *if there is no adversary that (t,ϵ)-breaks Π in that sense.*

5 Security

This section shows the concrete security reduction.

5.1 Basic Conversion

Lemma 10. (Chosen-Plaintext Security) *Suppose Π^{asym} is (t_1,ϵ_1)-secure in the sense of* OWE *and Π^{sym} is (t_2,ϵ_2)-secure in the sense of* FG. *Let l_1 and l_2 be the sizes of $\mathtt{MSPC}^{\mathrm{asym}}$ and $\mathtt{MSPC}^{\mathrm{sym}}$, respectively. Then Π^{hy} is (t,q_g,q_h,ϵ_0)-secure in the sense of* IND-CPA *in the random oracle model, where*

$$t = \min(t_1,t_2) - O(l_1+l_2) \quad \textit{and} \quad \epsilon_0 = 2(q_g+q_h)\epsilon_1 + \epsilon_2.$$

The proof is described in Appendix.

Lemma 11. (Knowledge Extractor) *Suppose* Π^{asym} *is* γ*-uniform and* (t_1, ϵ_1)*-secure in the sense of* OWE. *Suppose* Π^{sym} *is* (t_2, ϵ_2)*-secure in the sense of* FG. *Let* l_1 *and* l_2 *be the sizes of* $\mathtt{MSPC}^{\mathrm{asym}}$ *and* $\mathtt{MSPC}^{\mathrm{sym}}$*, respectively. Suppose* B *is a* (q_g, q_h, q_e)*-adversary for* Π^{hy}*. Then, there exist a* (t, λ)*-knowledge extractor,* K*, for* B *such that*

$$t = O((q_g + q_h) \cdot (l_1 + l_2)) \quad \textit{and} \quad \lambda = 1 - q_e \cdot \epsilon_1 - 2\epsilon_2 - \gamma - 2^{-l_2}.$$

The proof is described in Appendix.

Next is our main theorem. We omit the proof here since, due to the result of [4], it is straightforwards, provided lemmas 10 and 11 hold true (see [4]) (The proof will be described in the full paper version).

Theorem 12. (Chosen-Ciphertext Security) *Suppose* Π^{asym} *is* γ*-uniform and* (t_1, ϵ_1)*-secure in the sense of* OWE. *Suppose* Π^{sym} *is* (t_2, ϵ_2)*-secure in the sense of* FG. *Let* l_1 *and* l_2 *be the sizes of* $\mathtt{MSPC}^{\mathrm{asym}}$ *and* $\mathtt{MSPC}^{\mathrm{sym}}$*, respectively. Then* Π^{hy} *is* $(t, q_g, q_h, q_d, \epsilon)$*-secure in the sense of* IND-CCA *in the random oracle model where*

$$t = \min(t_1, t_2) - O((q_g + q_h) \cdot (l_1 + l_2)) \quad \textit{and}$$
$$\epsilon = (2(q_g + q_h)\epsilon_1 + \epsilon_2 + 1)(1 - 2\epsilon_1 - 2\epsilon_2 - \gamma - 2^{-l_2})^{-q_d} - 1.$$

5.2 A Variant: Symmetric Encryption is One-Time Padding

When a symmetric encryption, Π^{sym}, is one-time padding, we can relax the security condition.

Here define a symmetric encryption scheme by $\mathcal{E}_a^{\mathrm{sym}}(m) = a \oplus m$ (and $\mathcal{D}_a^{\mathrm{sym}}(c) = a \oplus c$). Define the key space $\mathtt{KSPC}^{\mathrm{sym}} = \{0,1\}^{l_2}$ and the message space $\mathtt{MSPC}^{\mathrm{sym}} = \{0,1\}^{l_2}$. Then $G : \mathtt{MSPC}^{\mathrm{asym}} \to \{0,1\}^{l_2}$ and $H : \mathtt{MSPC}^{\mathrm{asym}} \times \{0,1\}^{l_2} \to \{0,1\}^{l_2}$.

In $\Pi^{\mathrm{hy}} = (\mathcal{K}^{\mathrm{hy}}, \mathcal{E}^{\mathrm{hy}}, \mathcal{D}^{\mathrm{hy}}, \mathtt{COINS}^{\mathrm{hy}}, \{0,1\}^{l_2})$, the encryption of a plaintext m is then

$$\mathcal{E}_{pk}^{\mathrm{hy}}(m) = \quad \mathcal{E}_{pk}^{\mathrm{asym}}(\sigma; H(\sigma, m)) \quad || \quad G(\sigma) \oplus m.$$

Then we can show the following results:

Corollary 13. (Knowledge Extractor) *Suppose* Π^{asym} *is* γ*-uniform. Let* l_1 *be the size of* $\mathtt{MSPC}^{\mathrm{asym}}$*. Suppose* B *is a* (q_g, q_h, q_e)*-adversary for* Π^{hy}*. Then, there exist a* (t, λ)*-knowledge extractor,* K*, for* B *such that*

$$t = O((q_g + q_h) \cdot (l_1 + l_2)) \quad \textit{and} \quad \lambda = 1 - \gamma - 2^{-l_2}.$$

Theorem 14. (Chosen-Ciphertext Security) *Suppose* Π^{asym} *is* γ*-uniform and* (t_1, ϵ_1)*-secure in the sense of* OWE. *Let* l_1 *be the size of* $\mathtt{MSPC}^{\mathrm{asym}}$*. Then* Π^{hy} *is* $(t, q_g, q_h, q_d, \epsilon)$*-secure in the sense of* IND-CCA *in the random oracle model where*

$$t = t_1 - O((q_g + q_h) \cdot (l_1 + l_2)) \quad \textit{and}$$
$$\epsilon = (2(q_g + q_h)\epsilon_1 + 1)(1 - \gamma - 2^{-l_2})^{-q_d} - 1.$$

These proofs will be written in the full paper version.

6 Implementation

6.1 Implementation for the ElGamal Encryption Scheme

Let G_q be an Abelian group of the order q, where the group law is expressed by addition. We assume that the Diffie-Hellman problem defined over the underlying group is difficult. Let $\mathbf{g}$ be a generator of G_q, and $(\mathbf{y}, x)$ denote a pair of a public and secret keys such that $\mathbf{y} = x \cdot \mathbf{g}$, where $x \in \mathbb{Z}/q\mathbb{Z}$. Let $[0, 1, \ldots, q-1] \subseteq \mathtt{MSPC}^{\mathrm{sym}}$ be an encoding of G_q. Let $\mathtt{hash}_1 : [0, 1, \ldots, q-1] \longrightarrow \mathtt{KSPC}^{\mathrm{sym}}$ and $\mathtt{hash}_2 : [0, 1, \ldots, q-1] \times \mathtt{MSPC}^{\mathrm{sym}} \longrightarrow [0, 1, \ldots, q-1]$ be hash functions.

Encryption $\mathcal{E}^{\mathrm{hy}}_{pk}(m)$	Decryption $\mathcal{D}^{\mathrm{hy}}_{sk}(\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3})$
$\sigma \leftarrow_R [0, 1, \ldots, q-1]$,	$\hat{\sigma} := \mathbf{c_1} - x \cdot \mathbf{c_2}$,
$\mathbf{c_1} := \sigma + \mathtt{hash}_2(\sigma, m) \cdot \mathbf{y}$,	$\hat{m} := D^{\mathrm{sym}}_{\mathtt{hash}_1(\hat{\sigma})}(\mathbf{c_3})$,
$\mathbf{c_2} := \mathtt{hash}_2(\sigma, m) \cdot \mathbf{g}$,	If $\mathbf{c_1} = \hat{\sigma} + \mathtt{hash}_2(\hat{\sigma}, \hat{m}) \cdot \mathbf{y}$
$\mathbf{c_3} := \mathcal{E}^{\mathrm{sym}}_{\mathtt{hash}_1(\sigma)}(m)$.	then $m := \hat{m}$ else $m := \perp$.
return $(\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3})$.	return m.

Note 15. Let $k = |q|$ be a security parameter. The ElGamal encryption primitive, associated with 1^k, is 2^{-k}-uniform.

For an application to the elliptic curve encryption system, see [15].

6.2 Implementation for the Okamoto-Uchiyama Scheme

Let $n = p^2 q$ be a large positive integer such that p and q are both primes of the same size, i.e., $|p| = |q| = k + 1$. Let $\mathbb{Z}/n\mathbb{Z}$ and $(\mathbb{Z}/n\mathbb{Z})^{\times}$ be the integer ring modulo n and the multiplicative group of $\mathbb{Z}/n\mathbb{Z}$. We assume that the factoring of n is difficult. For $g, h_0 \in (\mathbb{Z}/n\mathbb{Z})^{\times}$, let $g_p = g^{p-1} \bmod p^2$ and let $h = h_0^n \bmod n$. Define $L(x) = \frac{x-1}{p}$ for $x \in \mathbb{Z}$. Let $\mathtt{hash}_1 : \{0,1\}^k \longrightarrow \mathtt{KSPC}^{\mathrm{sym}}$ and $\mathtt{hash}_2 : \{0,1\}^k \times \mathtt{MSPC}^{\mathrm{sym}} \longrightarrow \{0,1\}^{3k}$ be hash functions. Let $pk = (n, g, h, k)$ be the public-key and let $sk = (p, q, g_p, L(g_p))$ be the secret-key.

Encryption $\mathcal{E}^{\mathrm{hy}}_{pk}(m)$	Decryption $\mathcal{D}^{\mathrm{hy}}_{sk}(c_1, c_2)$
$\sigma \leftarrow_R \{0,1\}^k$,	$c_{1,p} := c_1^{p-1} \bmod p^2$,
$c_1 := g^{\sigma} h^{\mathtt{hash}_2(\sigma, m)} \bmod n$,	$\hat{\sigma} := L(c_{1,p}) \cdot L(g_p)^{-1} \bmod p$,
$c_2 := \mathcal{E}^{\mathrm{sym}}_{\mathtt{hash}_1(\sigma)}(m)$.	$\hat{m} := D^{\mathrm{sym}}_{\mathtt{hash}_1(\hat{\sigma})}(c_2)$,
	If $c_1 = g^{\hat{\sigma}} h^{\mathtt{hash}_2(\hat{\sigma}, \hat{m})} \pmod n$
	then $m := \hat{m}$ else $m := \perp$.
return (c_1, c_2).	return m.

Note 16. This Okamoto-Uchiyama encryption primitive, associated with 1^k, is 2^{-2k}-uniform ($2^{2k} \approx \phi(pq)$).

For more detailed information of this implementation, see [16].

7 Notes on γ-uniformity

As described in theorem 12, the security (IND-CCA) of our proposed scheme depends only on the security of the asymmetric and symmetric encryption primitives (OWE and FG, respectively) and γ-uniformity of the asymmetric encryption primitive. As γ increases to 1 (the variants of the encryption decrease), the security parameter ϵ, described in theorem 12, become larger (become worse). Since γ is evaluated at the worst point in the message space, one should choose $\mathtt{MSPC}^{\mathrm{asym}}$ very carefully, not including a singular point at which γ gets very close to 1. If $\gamma = 1$, then ϵ doesn't make sense any more (e.g., the asymmetric encryption primitive is a deterministic encryption scheme). Thus this conversion can't directly apply to such an asymmetric encryption scheme as the RSA encryption primitive. However, one can easily modify it and decrease parameter γ, as follows:

$$\hat{\mathcal{E}}_{pk}^{\mathrm{asym}}(m; (r||r')) = \mathcal{E}_{pk}^{\mathrm{asym}}(m; r) \;\; || \;\; r'$$

where $\mathcal{E}_{pk}^{\mathrm{asym}}(m; r)$ is an encryption algorithm in an asymmetric encryption scheme. Clearly the new asymmetric encryption scheme still meets the security notion of OWE if the original one meets the security notion. In particular, suppose the asymmetric encryption primitive is a OWTP and the symmetric encryption primitive is one-time padding. We then have, for a OWTP, f,

$$\mathcal{E}_{pk}^{\mathrm{hy}}(m) = (f(\sigma)||H(\sigma, m)) \;\; || \;\; (G(\sigma) \oplus m).$$

This coincides with the encryption scheme presented in [2] as a chosen-ciphertext secure encryption scheme (IND-CCA).

Acknowledgment

We would like to greatly thank Phil Rogaway for his invaluable support in revising our manuscript. We also wish to thank anonymous reviewers for useful comments.

References

1. M. Abdalla, M. Bellare and P. Rogaway, "DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem," Submission to IEEE P1363.
2. 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.
3. M. Bellare and P. Rogaway, "Optimal Asymmetric Encryption—How to encrypt with RSA" Advances in Cryptology –EUROCRYPT'94.
4. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations Among Notions of Security for Public-Key Encryption Schemes" Advances in Cryptology –CRYPTO'98.

5. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation", Proceedings of FOCS97, IEEE, 1997.
6. 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).
7. Canetti, R., Goldreich, O. and Halevi, S.: The Random Oracle Methodology, Revisited, Proc. of STOC, ACM Press, pp.209–218 (1998).
8. 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.
9. I. Damgård, "Towards practical public key systems secure against chosen ciphertext attacks", Advances in Cryptology –CRYPTO'91, pp.445–456, Proceedings, Lecture Notes in Computer Science No. 576, Springer-Verlag, 1992.
10. D. Dolev and C. Dwork and M. Naor, "Non-malleable cryptography", Proceeding of STOC91, pp 542–552.
11. 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.
12. Fujisaki, E. and Okamoto, T.: How to Enhance the Security of Public-Key Encryption at Minimum Cost, Proc.of PKC'99, LNCS, Springer-Verlag (1999).
13. S. Goldwasser, and S. Micali, "Probabilistic Encryption", JCSS, vol.28, pp.270–299, 1984.
14. M. Naor, and M. Yung "Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks", Proceeding of the 22nd Annual Symposium on Theory of Computing, ACM (STOC), pp.427–437, 1990
15. T. Okamoto, E. Fujisaki and H. Morita, "PSEC: Provably Secure Elliptic Curve Encryption Scheme", Submission to IEEE P1363a, March 1999.
16. T. Okamoto, S. Uchiyama and E. Fujisaki, "EPOC: Efficient Probabilistic Public-Key Encryption", Submission to IEEE P1363a, November 1998.
17. T. Okamoto, and S. Uchiyama, "A New Public-Key Cryptosystem as Secure as Factoring", Advances in Cryptology –EUROCRYPT'98, Springer-Verlag, 1998.
18. M. Rabin, "Digitalized Signatures and Public-Key Functions as Intractable as Factorization", MIT Technical Report, MIT/LCS/TR-212, 1979.
19. C. Rackoff and D.R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack", Advances in Cryptology –CRYPTO91, pp.433–444, Proceedings, Lecture Notes in Computer Science No. 576, Springer-Verlag, 1992.
20. 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.
21. V. Shoup, and R. Gennaro, "Securing Threshold Cryptosystems against Chosen Ciphertext Attack", Advances in Cryptology –EUROCRYPT'98, Springer-Verlag, 1998.
22. Y. Tsiounis and M. Yung, "On the Security of ElGamal based Encryption", PKC'98, January, 1998.
23. Y. Zheng and J. Seberry, "Practical Approaches to Attaining Security Against Adaptively Chosen Ciphertext Attacks", Advances in Cryptology –CRYPTO'92, pp.292–304, Proceedings, Lecture Notes in Computer Science No. 740, Springer-Verlag, 1992.

A Proof of Lemma 10

Suppose for contradiction that there exists an adversary, A_0, that $(t, q_g, q_h, \epsilon_0)$-breaks Π^{hy} in $\mathtt{MSPC}^{\mathrm{asym}}$ in the sense of IND-CPA in the random oracle model. We can then show that there exist adversaries, B_1 and B_2, such that B_1 (t_1, ϵ_1)-breaks Π^{asym} in $\mathtt{MSPC}^{\mathrm{asym}}$ in the sense of OWF and B_2 (t_2, ϵ_2)-breaks Π^{sym} in the sense of FG, where $t_1 = t + O(l_1 + l_2)$, $t_2 = t + O(l_1 + l_2)$, and $\epsilon_0 \leq 2 \cdot (q_g + q_h) \cdot \epsilon_1 + \epsilon_2$.

We show how to construct B_1 and B_2, which take advantage of A_0 as an oracle. Here, when utilizing A_0 as an oracle, B_1 and B_2 should make, by themselves, the answers for A_0's queries instead of random oracles, G and H, i.e., B_1 and B_2 have to simulate the random oracles. We describe the procedure, which is shared with B_1 and B_2, of the simulation in the following.

[**The procedure of making $\mathcal{T}_G$ and $\mathcal{T}_H$**] Recall that G and H are ideal random functions specified by $G : \mathtt{MSPC}^{\mathrm{asym}} \to \mathtt{KSPC}^{\mathrm{sym}}$, and $H : \mathtt{MSPC}^{\mathrm{asym}} \times \mathtt{MSPC}^{\mathrm{sym}} \to \mathtt{COINS}^{\mathrm{asym}}$. At first we prepare empty lists, $\mathcal{T}_G$ and $\mathcal{T}_H$ and a counter, $\mathtt{count} \leftarrow 0$.

How to make $\mathcal{T}_G$ For a query, σ, if it hasn't been entered as an entry in $\mathcal{T}_G$, choose g random and uniformly from $\mathtt{KSPC}^{\mathrm{sym}}$, answer g to adversary A_0, increase $\mathtt{count}$ by 1, set $(\sigma_{\mathtt{count}}, g_{\mathtt{count}}) := (\sigma, g)$, and put it on $\mathcal{T}_G$, otherwise answer g_i such that $\sigma_i = \sigma$ and $(\sigma_i, g_i) \in \mathcal{T}_G$.

How to make $\mathcal{T}_H$ For a query, (σ, m), if it hasn't been entered as an entry in $\mathcal{T}_H$, choose h random and uniformly from $\mathtt{COINS}^{\mathrm{asym}}$, answer h to adversary A_0, increase $\mathtt{count}$ by 1, set $(\sigma_{\mathtt{count}}, m_{\mathtt{count}}, h_{\mathtt{count}}) := (\sigma, m, h)$, and put it on $\mathcal{T}_H$, otherwise answer h_i such that $(\sigma_i, m_i) = (\sigma, m)$ and $(\sigma_i, m_i, g_i) \in \mathcal{T}_H$.

[**Adversary B_1**] We explain the specification of adversary B_1. Recall that B_1 is an algorithm that on input (pk, y) outputs some string, and the advantage is specified by $Adv^{\mathrm{cpa}}_{B_1, \Pi^{\mathrm{asym}}}(k) = \Pr[(pk, sk) \leftarrow \mathcal{K}^{\mathrm{asym}}(1^k); x \leftarrow \mathtt{MSPC}^{\mathrm{asym}}; y \leftarrow \mathcal{E}^{\mathrm{asym}}_{pk}(x) : B_1(pk, y) = \mathcal{D}^{\mathrm{asym}}_{sk}(y)]$. B_1 runs A_0 as follows:

Step 1 Input pk to A_0 (originally inputted to B_1) and run A_0 in the find mode. If A_0 asks oracles, G and H, then follow the above procedure. Finally A_0 would output (x_0, x_1, s).

Step 2 Choose $b \leftarrow_R \{0, 1\}$ and $\hat{g} \leftarrow_R \mathtt{KSPC}^{\mathrm{sym}}$. Then set $c_1 := y$ and $c_2 := \mathcal{E}^{\mathrm{sym}}_{\hat{g}}(x_b)$.

Step 3 Input $(x_0, x_1, s, (c_1, c_2))$ to A_0 and run A_0 in the guess mode. If A_0 asks oracles, G and H, then follow the above procedure. After asking at most $(q_g + q_h)$ queries to the random oracles or running in at most time t, A_0 would output bit b'. However if it is still running, abort it.

Step 4 Choose a number $i \leftarrow_R \{1, \ldots, \mathtt{count}\}$ and output σ_i as the answer to y (originally inputted to B_1).

[**Adversary B_2**] We now describe the specification of adversary B_2. Recall that B_2 is an algorithm that has two modes: At first in the find mode it executes and outputs two distinct messages and some information used

in the next mode. In the guess mode, it runs on input ciphertext as well as the above information, and outputs one bit. The advantage is specified by $Adv^{\mathrm{fg}}_{B_2,\Pi^{\mathrm{sym}}}(k) = 2 \cdot \Pr[a \leftarrow \mathtt{KSPC}^{\mathrm{sym}}; (x_0, x_1, s) \leftarrow B_2(\mathrm{find}); b \leftarrow_R \{0,1\}; y = \mathcal{E}^{\mathrm{sym}}_a(x_b) : B_2(\mathrm{guess}, x_0, x_1, s, y) = b] - 1$. B_2 runs A_0 as follows:

Step 1 (B_2 is set in the find mode.) Run $\mathcal{K}^{\mathrm{asym}}$ on input 1^k and let K output (pk, sk). Do the same thing as in the first step in the case of adversary B_1. Finally A_0 outputs (x_0, x_1, s), then output (x_0, x_1, s) as his own output.

Step 2 (B_2 is inputted $y = \mathcal{E}^{\mathrm{sym}}_a(x_b)$ where $a \leftarrow_R \mathtt{KSPC}^{\mathrm{sym}}$, $b \leftarrow_R \{0,1\}$, and enter the guess mode.) Choose $\sigma \leftarrow_R \mathtt{MSPC}^{\mathrm{asym}}$ and $\hat{h} \leftarrow_R \mathtt{COINS}^{\mathrm{asym}}$. Then set $c_1 := \mathcal{E}^{\mathrm{asym}}(\sigma, \hat{h})$ and $c_2 := y$.

Step 3 (B_2 is still in the guess mode.) Do the same thing as in the third step in the case of adversary B_1.

Step 4 (B_2 is still in the guess mode.) Finally, A_0 outputs b'. Then output b' as his own answer.

From the specifications of B_1 and B_2, their running times are $t + O(l_1 + l_2)$. Here we define, for shorthand, the following experiment:

$$\begin{aligned}
\mathrm{Ask}A_0 &= [\ A_0 \text{ asks } G \text{ or } H \text{ a query that includes } \mathcal{D}^{\mathrm{asym}}_{sk}(c_1).] \\
\mathrm{Succ}A_0 &= [G, H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}^{\mathrm{asym}}(1^k); (x_0, x_1, s) \leftarrow {A_0}^{G,H}(\mathrm{find}, pk); \\
&\quad b \leftarrow_R \{0,1\}; y \leftarrow \mathcal{E}^{\mathrm{hy}}_{pk}(x_b) : {A_0}^{G,H}(\mathrm{guess}, x_0, x_1, s, y) = b] \\
\mathrm{Succ}B_1 &= [(pk, sk) \leftarrow \mathcal{K}^{\mathrm{asym}}(1^k); x \leftarrow \mathtt{MSPC}^{\mathrm{asym}}; y \leftarrow \mathcal{E}^{\mathrm{asym}}_{pk}(x) : \\
&\quad B_1(pk, y) = \mathcal{D}^{\mathrm{asym}}_{sk}(y)] \\
\mathrm{Succ}B_2 &= [sk_2 \leftarrow_R \mathtt{KSPC}^{\mathrm{sym}}; (x_0, x_1, s) \leftarrow B_2(\mathrm{find}); b \leftarrow_R \{0,1\}; \\
&\quad y = \mathcal{E}^{\mathrm{sym}}_a(x_b) : {A_0}^{G,H}(\mathrm{guess}, x_0, x_1, s, y) = b]
\end{aligned}$$

In addition, let $p_0 = \Pr[\mathrm{Ask}A_0]$, so we can write

$$\begin{aligned}
\Pr[\mathrm{Succ}A_0] &= \Pr[\mathrm{Succ}A_0 | \mathrm{Ask}A_0] \cdot p_0 + \Pr[\mathrm{Succ}A_0 | \overline{\mathrm{Ask}A_0}] \cdot (1 - p_0), \\
\Pr[\mathrm{Succ}B_2] &= \Pr[\mathrm{Succ}B_2 | \mathrm{Ask}A_0] \cdot p_0 + \Pr[\mathrm{Succ}B_2 | \overline{\mathrm{Ask}A_0}] \cdot (1 - p_0).
\end{aligned}$$

Then, from the specification of adversaries, B_1 and B_2, it holds

$$\Pr[\mathrm{Succ}B_1] \geq (q_g + q_h)^{-1} \cdot p_0 \text{ and } \Pr[\mathrm{Succ}B_2] \geq \Pr[\mathrm{Succ}A_2 | \overline{\mathrm{Ask}A_0}] \cdot (1 - p_0).$$

This is because: if A_0 asks at least one query including $\mathcal{D}^{\mathrm{asym}}_{sk}(y)$ to either G or H, then B_1 can output the correct answer with probability at least $1/(q_g + q_h)$. Otherwise, $\Pr[\mathrm{Succ}A_0 | \overline{\mathrm{Ask}A_0}] \cdot (1 - p_0) = \Pr[\mathrm{Succ}B_2 | \overline{\mathrm{Ask}A_0}] \cdot (1 - p_0)$. Therefore,

$$\Pr[\mathrm{Succ}A_0] \leq (q_g + q_h) \cdot \Pr[\mathrm{Succ}B_1] + \Pr[\mathrm{Succ}B_2]. \tag{1}$$

Then, from the assumption (for contradiction), we can write

$$\epsilon_0 \leq 2 \Pr[\mathrm{Succ}A_0] - 1, \quad \epsilon_1 = \Pr[\mathrm{Succ}B_1], \quad \epsilon_2 = 2 \Pr[\mathrm{Succ}B_2] - 1. \tag{2}$$

Hence, $\epsilon_0 \leq 2(q_g + q_h) \cdot \epsilon_1 + \epsilon_2$. □

B Proof of Lemma 11

Let B be a (q_g, q_h, q_e)-adversary that, on input pk, asks queries to G and H, asks queries to the encryption oracle, $\mathcal{E}^{\mathrm{hy}}_{pk}(\cdot)$, and finally outputs (c_1, c_2), where $(c_1, c_2) \notin \mathcal{Y}$. Recall we write the experiment as $(\mathcal{T}_G, \mathcal{T}_H, \mathcal{Y}, y) \leftarrow B^{G,H,\mathcal{E}^{\mathrm{hy}}_{pk}}(pk)$. The knowledge extractor, K, is an algorithm which, on input $(\mathcal{T}_G, \mathcal{T}_H, \mathcal{Y}, (c_1, c_2), pk)$, outputs a string. Recall for $k \in \mathbb{N}$, $\mathrm{Succ}^{\mathrm{ke}}_{K,B,\Pi^{\mathrm{hy}}}(k) =$

$$\Pr[G, H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}(1^k); (\mathcal{T}_G, \mathcal{T}_H, \mathcal{Y}, y) \leftarrow B^{H,\mathcal{E}^{\mathrm{hy}}_{pk}}(pk) : K(\mathcal{T}_G, \mathcal{T}_H, \mathcal{Y}, y, pk) = \mathcal{D}^{\mathrm{hy}}_{sk}(y)].$$

[Knowledge Extractor] Here, let $\mathcal{T}_G = \{(\sigma_i, g_i) | i = 1, \ldots, q_g\}$ and $\mathcal{T}_H = \{(\sigma'_j, m_j, h_j) | j = 1, \ldots, q_h\}$. We then give the specification of the knowledge extractor K as follows:

Step 1 Set two empty lists, S_1 and S_2.

Step 2 Find all elements in $\mathcal{T}_H$ such that $c_1 = \mathcal{E}^{\mathrm{asym}}_{pk}(\sigma'_j, h_j)$ and put them into list S_1. If $S_1 = \emptyset$, then output $\perp$, otherwise

Step 3 For every (σ'_j, m_j, h_j) in S_1, find all elements in $\mathcal{T}_G$ such that $\sigma_i = \sigma'_j$ and put them (i.e., $(\sigma'_j, m_j, h_j)||(\sigma_i, g_i)$'s) into S_2. If $S_2 = \emptyset$, then output $\perp$, otherwise

Step 4 Check in S_2 if there exists a $(\sigma'_j, m_j, h_j)||(\sigma_i, g_i)$ such that $c_2 = \mathcal{E}^{\mathrm{sym}}_{g_i}(m_j)$. If it exists in S_2, then output m_j otherwise output $\perp$.

This protocol runs in time $O((q_g + q_h)k)$.

Next we examine the advantage of the knowledge extractor. We define the following events:

- $\mathtt{Inv}$ is true if there exists $(c_1^*, c_2^*) \in \mathcal{Y}$ and $(\sigma_i, g_i) \in \mathcal{T}_G$ or $(\sigma_j, m_j, h_j) \in \mathcal{T}_H$ such that $\sigma_i = D^{\mathrm{asym}}_{sk}(c_1^*)$ or $\sigma_j = D^{\mathrm{asym}}_{sk}(c_1^*)$.
- $p(S_1)$ is true if $S_1 \neq \emptyset$.
- $p(S_2)$ is true if $S_2 \neq \emptyset$.
- $\mathtt{Find}$ is true if there exists a $(\sigma'_j, m_j, h_j)||(\sigma_i, g_i)$ in S_2 such that $c_2 = \mathcal{E}^{\mathrm{sym}}_{g_i}(m_j)$.
- $\mathtt{Fail}$ is true if "the output of knowledge extractor K" $\neq \mathcal{D}^{\mathrm{hy}}_{sk}(c_1, c_2)$.

We further define the following events:

$$\begin{aligned}
\text{`}\mathtt{1}\text{'} &= \mathtt{Inv}.\\
\text{`}\mathtt{00}\text{'} &= \neg\mathtt{Inv} \wedge \neg p(S_1).\\
\text{`}\mathtt{010}\text{'} &= \neg\mathtt{Inv} \wedge p(S_1) \wedge \neg p(S_2).\\
\text{`}\mathtt{0110}\text{'} &= \neg\mathtt{Inv} \wedge p(S_1) \wedge p(S_2) \wedge \neg\mathtt{Find}.\\
\text{`}\mathtt{0111}\text{'} &= \neg\mathtt{Inv} \wedge p(S_1) \wedge p(S_2) \wedge \mathtt{Find}.
\end{aligned}$$

Then the following equation holds

$$\begin{aligned}
\Pr[\mathtt{Fail}] = \Pr[\mathtt{Fail}|\mathtt{1}] \cdot \Pr[\mathtt{1}] + \Pr[\mathtt{Fail}|\mathtt{00}] \cdot \Pr[\mathtt{00}] + \Pr[\mathtt{Fail}|\mathtt{010}] \cdot \Pr[\mathtt{010}] +\\
\Pr[\mathtt{Fail}|\mathtt{0110}] \cdot \Pr[\mathtt{0110}] + \Pr[\mathtt{Fail}|\mathtt{0111}] \cdot \Pr[\mathtt{0111}].
\end{aligned}$$

Hence,

$$\Pr[\mathrm{Fail}] \leq \Pr[\mathtt{1}] + \Pr[\mathrm{Fail}|\mathtt{00}] + \Pr[\mathrm{Fail}|\mathtt{010}] + \Pr[\mathrm{Fail}|\mathtt{0110}] + \Pr[\mathrm{Fail}|\mathtt{0111}].$$

Here we can easily find that $\Pr[\mathrm{Fail}|\mathtt{0110}] = \Pr[\mathrm{Fail}|\mathtt{0111}] = 0$. In addition, we claim the following inequalities hold true:

Claim. $\Pr[\mathtt{1}] \leq q_e \cdot \epsilon_1$.

Proof. Remember that the interaction between adversary B and encryption oracle $\mathcal{E}^{\mathrm{hy}}_{pk}(\cdot)$: When B make access to $\mathcal{E}^{G,H}_{pk}$ with query m, $\mathcal{E}^{\mathrm{hy}}_{pk}$ takes random coins $\sigma \leftarrow_R \mathtt{COINS}^{\mathrm{asym}}$ and answer to B with $(\mathcal{E}^{\mathrm{asym}}_{pk}(\sigma, H(\sigma, m)) \;\;||\;\; \mathcal{E}^{\mathrm{sym}}_{G(\sigma)}(m))$. B makes at most q_e queries to $\mathcal{E}^{\mathrm{hy}}_{pk}$.

Therefore, $\Pr[\mathtt{1}] = \Pr[\mathrm{Inv}] \leq q_e \cdot \epsilon_1$. □

Claim. $\Pr[\mathrm{Fail}|\mathtt{00}] \leq \gamma$

Proof. Given the event 00, we can identify B with an adversary B' which on input pk outputs a string, y, to guess the random coins in y. The advantage of B' is $\mathrm{Adv}_{B',\Pi^{\mathrm{asym}}}(k) =$

$$\Pr[(pk, sk) \leftarrow \mathcal{K}^{\mathrm{asym}}(1^k); h \leftarrow_R \mathtt{COINS}^{\mathrm{asym}}; y \leftarrow B'(pk); \\ x \leftarrow \mathcal{D}^{\mathrm{asym}}_{sk}(y) : y = \mathcal{E}^{\mathrm{asym}}_{pk}(x; h)]$$

Then

$$\begin{aligned}\Pr[\mathrm{Fail}|\mathtt{00}] &\leq \mathrm{Adv}_{B',\Pi^{\mathrm{asym}}}(k) \\ &= \Pr[(pk, sk) \leftarrow \mathcal{K}^{\mathrm{asym}}(1^k); y \leftarrow B'(pk); x \leftarrow \mathcal{D}^{\mathrm{asym}}_{sk}(y); \\ &\quad h \leftarrow_R \mathtt{COINS}^{\mathrm{asym}} : y = \mathcal{E}^{\mathrm{asym}}_{pk}(x; h)].\end{aligned}$$

Recall, for $(pk, sk) \in \mathcal{K}^{\mathrm{asym}}(1^k)$, $x \in \mathtt{MSPC}^{\mathrm{asym}}$ and $y \in \{0,1\}^*$,

$$\gamma(x, y) = \Pr[h \leftarrow_R \mathtt{COINS}^{\mathrm{asym}} : y = \mathcal{E}^{\mathrm{asym}}_{pk}(x; h)] \leq \gamma.$$

Hence $\Pr[\mathrm{Fail}|\mathtt{00}] \leq \gamma$. □

Claim. $\Pr[\mathrm{Fail}|\mathtt{010}] \leq 2\epsilon_2 + 2^{-l_2}$.

Proof. Given the event 010, we can identify B with an adversary B' that outputs a pair of strings to guess the secret-key of Π^{sym}. The advantage of B' is $\mathrm{Adv}_{B',\Pi^{\mathrm{sym}}}(k) =$

$$\Pr[g \leftarrow_R \mathtt{KSPC}^{\mathrm{sym}}; (x, y) \leftarrow B'(\mathrm{find}) : y = \mathcal{E}^{\mathrm{sym}}_g(x)].$$

Since the event, Fail|010, means that B' outputs valid (x, y) (extractor K outputs $\perp$), $\Pr[\mathrm{Fail}|\mathtt{010}] = \mathrm{Adv}_{B',\Pi^{\mathrm{sym}}}(k)$.

[**Adversary** A] Suppose there exists B' with $\delta := \mathrm{Adv}_{B',\Pi^{\mathrm{sym}}}(k)$. We then construct adversary A against Π^{sym} as follows:

Step 1 (A starts in the find mode.) Run B' and finally B' outputs (x', y').

Step 2 Choose $b' \leftarrow_R \{0,1\}$ and $x'' \leftarrow_R \{0,1\}^{|x'|}$. Then set $x_{b'} := x'$ and $x_{\bar{b'}} := x''$ (where $\bar{b'}$ denotes the complement of b'). Finally output (x_0, x_1).

Step 3 (A is inputted $y = \mathcal{E}_g^{\mathrm{sym}}(x_b)$ where $g \leftarrow_R \mathtt{KSPC}^{\mathrm{sym}}$ and $b \leftarrow_R \{0,1\}$ and enters the guess mode.) If $y = y'$ then output b' else flip a coin again and output the result (namely, after $b'' \leftarrow_R \{0,1\}$ output b'').

Define $\mathrm{Succ}A =$

$$[g \leftarrow_R \mathtt{KSPC}^{\mathrm{sym}}; (x_0, x_1) \leftarrow A(\mathrm{find}); b \leftarrow \{0,1\}; y = \mathcal{E}_g^{\mathrm{sym}}(x_b) : A(\mathrm{guess}, s, y) = b].$$

Then,

$$\Pr[\mathrm{Succ}A] = \Pr[\mathrm{Succ}A | y = y'] \cdot \Pr[y = y'] + \Pr[\mathrm{Succ}A | y \neq y'] \cdot \Pr[y \neq y'].$$

Define $X = (x, y)$, $Q_X = \Pr_B[(x, y) \leftarrow B'(\mathrm{find}]$, and $P_X = \frac{\#\{g | y = \mathcal{E}_g^{\mathrm{sym}}(x)\}}{\#\mathtt{KSPC}^{\mathrm{sym}}}$. We then have

$$\delta = \sum_X Q_X P_X.$$

To evaluate $\Pr[\mathrm{Succ}A]$, let $X' := (x', y')$ and $X'' := (x'', y')$. Here note that $Q_{X'} = Q_{X''}$. First we evaluate the conditional probability of $\Pr[\mathrm{Succ}A]$ given X', X'' (outputs of B and A); we write the probability as $\Pr_{X',X''}[\mathrm{Succ}A]$.

$$\Pr_{X',X''}[y = y'] = \frac{1}{2} \cdot \Pr_{X',X''}[y' = \mathcal{E}_g^{\mathrm{sym}}(x')] + \frac{1}{2} \cdot \Pr_{X',X''}[y' = \mathcal{E}_g^{\mathrm{sym}}(x'')]$$

$$= \frac{1}{2}(P_{X'} + P_{X''}),$$

$$\Pr_{X',X''}[\mathrm{Succ}A | y = y'] = \frac{\#\{g \mid y' = \mathcal{E}_g^{\mathrm{sym}}(x')\}}{\#\{g \mid y' = \mathcal{E}_g^{\mathrm{sym}}(x') \vee y' = \mathcal{E}_g^{\mathrm{sym}}(x'')\}} = \frac{P_{X'}}{P_{X'} + P_{X''}},$$

$\Pr_{X',X''}[\mathrm{Succ}A | y \neq y'] = \frac{1}{2}$, and $\Pr_{X',X''}[y \neq y'] = 1 - \Pr_{X',X''}[y = y'] = 1 - \frac{1}{2}(P_{X'} + P_{X''})$. Therefore,

$$\Pr_{X',X''}[\mathrm{Succ}A] = \frac{1}{2} + \frac{1}{4}(P_{X'} - P_{X''}).$$

We then evaluate $\Pr[\mathrm{Succ}A]$ (Here note that $Q_{X'}$ is taken over B''s coin flips and $\Pr[x'' \leftarrow A(\mathrm{find})] = 2^{-l_2}$ is taken over A's coin flips.)

$$\Pr[\mathrm{Succ}A] = \left(\frac{1}{2}\right)^{l_2} \sum_{X'} \sum_{X''} Q_{X'}(\frac{1}{2} + \frac{1}{4}(P_{X'} - P_{X''}))$$

$$= \frac{1}{2} + \frac{1}{4}\delta - \frac{1}{4} \sum_{X'} \sum_{X''} Q_{X'} P_{X''},$$

since $\sum_{X'} Q_{X'} P_{X'} = \delta$, $\sum_{X'} Q_{X'} = 1$, and $\sum_{x''} \left(\frac{1}{2}\right)^{l_2} = 1$.

By combining with the assumption that Π^{sym} is (t_2, ϵ_2)-secure, i.e., $\Pr[\mathrm{Succ}A] \leq \frac{1}{2} + \frac{\epsilon_2}{2}$ (that is, $\frac{1}{2} + \frac{1}{4}\delta - \frac{1}{4}\sum_{X'}\sum_{X''} Q_{X'} P_{X''} \leq \frac{1}{2} + \frac{\epsilon_2}{2}$), we have the following inequality,

$$\delta \leq 2\epsilon_2 + \sum_{X'} \sum_{X''} Q_{X'} P_{X''},$$

We then evaluate $\sum_{X'} \sum_{X''} Q_{X'} P_{X''}$.

$$\sum_{X'} \sum_{X''} Q_{X'} P_{X''} = \sum_{X'} Q_{X'} \Big(\Big(\frac{1}{2}\Big)^{l_2} \sum_{x''} \frac{\#\{g \mid y' = \mathcal{E}_g^{\mathrm{sym}}(x'')\}}{\#\mathtt{KSPC}^{\mathrm{sym}}} \Big).$$

Since $\{g \mid y' = \mathcal{E}_g^{\mathrm{sym}}(x_1'')\}$ should be disjoint from $\{g \mid y' = \mathcal{E}_g^{\mathrm{sym}}(x_2'')\}$ (with $x_1'' \neq x_2''$) in order to uniquely decrypt y' with key g, $\sum_{x''} \#\{g \mid y' = \mathcal{E}_g^{\mathrm{sym}}(x'')\} \leq \#\mathtt{KSPC}^{\mathrm{sym}}$. Hence,

$$\sum_{X'} \sum_{X''} Q_{X'} P_{X''} \leq \Big(\frac{1}{2}\Big)^{l_2} \sum_{X'} Q_{X'} = \Big(\frac{1}{2}\Big)^{l_2}.$$

Thus, $(\delta =)\mathrm{Adv}_{B', \Pi^{\mathrm{sym}}}(k) \leq 2\epsilon_2 + 2^{-l_2}$. □

From the claims above, $\Pr[\mathrm{Fail}] \leq q_e \cdot \epsilon_1 + \gamma + 2\epsilon_2 + 2^{-l_2}$.

Therefore,

$$\lambda = 1 - \Pr[\mathrm{Fail}] \geq 1 - (q_e \cdot \epsilon_1 + 2\epsilon_2 + 2^{-l_2} + \gamma).$$

□