J. Cryptol. (2013) 26: 80–101
DOI: 10.1007/s00145-011-9114-1

Journal of
CRYPTOLOGY

Secure Integration of Asymmetric and Symmetric Encryption Schemes*

Eiichiro Fujisaki and Tatsuaki Okamoto
NTT Laboratories, 3-9-11 Midori-cho Musashino-shi, Tokyo 180-8585, Japan
fujisaki.eiichiro@lab.ntt.co.jp; okamoto.tatsuaki@lab.ntt.co.jp

Communicated by Dan Boneh

Received 20 April 2005
Online publication 2 December 2011

Abstract. This paper presents a generic conversion from weak asymmetric and symmetric encryption schemes to an asymmetric encryption scheme that is chosen-ciphertext secure in the random oracle model. Our conversion is the first generic transformation from an arbitrary one-way asymmetric encryption scheme to a chosen-ciphertext secure asymmetric encryption scheme in the random oracle model.

Key words. Asymmetric and symmetric (or public-key and private-key) encryptions, Generic conversion, Indistinguishability against chosen ciphertext attacks (IND-CCA), Random oracle model, Security proof.

1. Introduction

Suppose that an asymmetric (aka public-key) encryption scheme is secure in a very weak sense—an adversary cannot entirely decrypt the encryption of a random plaintext, called one-wayness. Suppose that a symmetric (aka private-key) encryption scheme is secure in the following weak sense—an adversary cannot distinguish the encryption of m_1 from the encryption of m_2, encrypted under a one-time private key, called one-time security. In addition, suppose that any plaintext in the message space of the asymmetric encryption scheme has at least $2^{(\log k)}$ possible ciphertexts, called $(\log k)$-spread. Given these encryption schemes, we construct a new hybrid encryption scheme. The (hybrid) encryption of message m is defined as

$$\mathcal{E}^{\mathsf{hy}}_{pk}(m;\sigma) = \mathcal{E}^{\mathsf{asy}}_{pk}\big(\sigma; H(\sigma, c)\big) \parallel \mathcal{E}^{\mathsf{sy}}_{G(\sigma)}(m),$$

where

- $\mathcal{E}^{\mathsf{asy}}_{pk}(\text{message; coins})$ indicates the asymmetric encryption of the indicated message using the indicated coins as random bits,

* This is the full version of the paper [18] by fixing bugs and providing a clean, formal proof associated with a better security bound.

- $\mathcal{E}^{\mathsf{sy}}_a$(message) indicates the symmetric encryption of the indicated message using the private key a,
- σ is a random string chosen from an appropriate domain,
- $c = \mathcal{E}^{\mathsf{sy}}_{G(\sigma)}(m)$, and
- G and H denote hash functions.

In the random oracle model (G and H are modeled as random oracles), this hybrid encryption scheme (with an appropriate decryption algorithm) is chosen-ciphertext secure, i.e., secure in the sense of indistinguishability against adaptive chosen-ciphertext attacks [40], called IND-CCA.

Any asymmetric encryption scheme can be converted to an asymmetric encryption scheme with $(\log k)$-spread, and a one-time secure symmetric encryption scheme can be constructed by extending private key a to a pseudo random string with the same length of message m (via a pseudo random generator) and xoring it with m. Therefore, the above transformation provides a generic conversion from any one-way asymmetric encryption scheme to an IND-CCA asymmetric encryption scheme.

1.1. *Security of Encryption Schemes*

The fundamental security we require for asymmetric and symmetric encryption schemes is semantic security—the asymmetric [24] and the symmetric [22] cases. Informally, it is measured by the infeasibility of an adversary to learn nothing more about the plaintext from the ciphertext than what is already known. Semantic security has an equivalent notion called indistinguishability [24], which is defined as the inability of the adversary to distinguish which of two messages is encrypted. We say that the adversary wins if it can correctly guess the encryption of two messages with a significant probability of more than a half.

Goldwasser and Micali [24] proved that indistinguishability of encryptions implies semantic security in the asymmetric case. Goldreich [22] proved the equivalence of two definitions in both asymmetric and symmetric cases.

We note that in the above notions the adversary is only allowed to take the encryption of a single message, which implies that the security guarantee of an encryption scheme is provided only when it is used to encrypt a single plaintext per generated key. In fact, the single-message indistinguishability does not imply the multiple-message one in the symmetric case. Namely, a symmetric encryption scheme secure only in the single-message sense cannot safely encrypt two messages under the same private key. Fortunately, the single-message security implies the multiple-message version in the asymmetric case [22]. In this paper, we use *one-time security* to refer to (the single-message) indistinguishability of symmetric encryptions defined in [22].

In the asymmetric case, indistinguishability is equivalent to indistinguishability against chosen plaintext attacks, which we call IND-CPA. Naor and Yung [33] proposed a stronger security notion of asymmetric encryptions, in which the adversary is additionally allowed to access the decryption oracle before the challenge ciphertext is given. This notion is called IND-CCA1 (or indistinguishability against a-priori chosen-ciphertext attacks). Rackoff and Simons [40] proposed a further stronger security notion, in which an adversary may access the decryption oracle even after the challenge ciphertext is given. The only constraint of the adversary is that it is not allowed to ask

for the decryption of the challenge ciphertext after it was given. This security notion is indistinguishability against (a-posteriori) chosen-ciphertext attacks, called IND-CCA2 or IND-CCA. IND-CCA is widely recognized as an appropriate requirement for public-key (or asymmetric) encryptions.

An asymmetric encryption scheme is one-way if any adversary cannot entirely decrypt the encryption of a random plaintext, but this notion is too weak to protect the privacy of messages because it only guarantees that $(\log k)$ bits of the plaintext is infeasible to determine.

1.2. *Contribution*

This paper provides a generic secure hybrid usage of arbitrary asymmetric and symmetric encryption schemes in the random oracle model. In particular, this is the first construction to convert any one-way public-key encryption scheme to an IND-CCA secure public-key encryption scheme in the random oracle model.

1.3. *Related Work*

1.3.1. *CCA Asymmetric Encryptions from General Assumptions*

Naor and Yung [33] first proposed IND-CCA1 public-key encryption schemes and Dolev, Dwork and Naor [16] then provided the first instantiation of IND-CCA2 (or IND-CCA) public-key encryption schemes. Both schemes are based on general assumptions (trap-door permutations) and need non-interactive zero knowledge proofs for proving two encryptions under independent, distinct public-keys implies the same plaintext. This work was later followed by Sahai [42] and Lindell [32] with *simulation-sound* non-interactive zero-knowledge proofs. These constructions are generic but very inefficient because it is expensive to construct such non-interactive zero-knowledge proofs in general.

1.3.2. *Generic Constructions in Random Oracle Model*

Early attempts at constructing practical IND-CCA public-key encryption schemes were done; Damgård (IND-CCA1 under a non-standard assumption) [15], followed by Zheng and Seberry [47] and Lim and Lee [31] (the scheme in [31] was cryptoanalyzed by Frankel and Yung [17]). The key idea of this approach to constructing these schemes is to create an encryption scheme in a way that the attacker cannot produce a valid ciphertext without knowing the plaintext. We note that these schemes are based on specific number-theoretic assumptions and do not support general assumptions.

Bellare and Rogaway formalized the above approach and introduced the notion of *plaintext-awareness*, which first appeared in [6]; but revised later in [7] for incompleteness. The former notion is called PA1, whereas the latter is called PA2. PA1 implies IND-CCA1, but not IND-CCA2, while PA2 implies IND-CCA2. Plaintext-awareness [6,7] was defined in the random oracle model [5],[1] which is a world where there is a public random function to which all parties (including the adversary) can make oracle access. We say that a cryptographic protocol designed in the real world is

[1] Later, Bellare and Palacio [4] defined plaintext-awareness in the standard model, but it is off topic.

secure in the random oracle model if it is secure when the hash functions plugged in the cryptographic protocol are replaced by the random oracle.

Bellare and Rogaway suggested (without providing a rigorous proof) a conversion from any trap-door permutation to an IND-CCA2 public-key encryption scheme in the random oracle model [5]. They also presented a method, called Optimal Asymmetric Encryption Padding (OAEP), that converts any trap-door permutation f into an IND-CCA1 public-key encryption scheme in the random oracle model [6]. Later, Shoup [44] reported that the OAEP conversion is *not sufficient* to provide IND-CCA2 security. Shoup proved that f-OAEP only meets PA1 for trap-door permutation f. He also illustrated a specific trap-door permutation f such that f-OAEP is not IND-CCA2. Fujisaki, Okamoto, Pointcheval, and Stern [20] showed a *sufficient condition* that OAEP provides IND-CCA2 security, where f-OAEP is IND-CCA2 if f meets partial one-wayness. They also proved that the RSA function is partial one-way under the RSA assumption. Thus, RSA-OAEP is IND-CCA2 in the random oracle model.

Although both schemes proposed in [5,6] support arbitrary trap-door permutations, they cannot be applied to public-key encryption schemes. We note that assuming the existence of (one-way) public-key encryptions is weaker than assuming that of trap-door permutations.

We presented a generic conversion from an arbitrary IND-CPA public-key encryption scheme (with a long enough message space) to an IND-CCA2 one in the random oracle model [19]. Although IND-CPA public-key encryptions can be constructed from one-way public-key encryptions by using hard core predicates [23], the direct construction from one-way public-key encryptions is more preferable in the sense of efficiency. Independently of us, Pointcheval [39] proposed a generic conversion from an arbitrary one-way public-key encryption scheme to an IND-CCA2 one in the random oracle model. His conversion is, however, slightly less efficient than our conversion. Okamoto and Pointcheval [34] presented another generic construction, but the starting public-key encryption schemes are required to be stronger than ours.

1.3.3. *Constructions in Standard Model*

Cramer and Shoup [11] presented the first practical IND-CCA public-key encryption scheme without random oracles under the decisional Diffie–Hellman (DDH) assumption. They then introduced hash proof systems [12,13], generalizing the heart of their design methodology. Hash proof systems have been implicitly or explicitly used or extended in many papers for designing practical IND-CCA public-key encryptions on specific number-theoretic assumptions, e.g., [10,14,25,26,29,30,46]. Boneh, Canetti, Halevi, and Katz [8,9] presented another pass by proposing generic conversions from a selective ID secure identity-based encryption scheme to an IND-CCA public-key encryption scheme. Peikert and Waters [37] and subsequently Rosen and Segev [41] recently proposed another method for designing IND-CCA public-key encryptions via so-called lossy/correlated trap-door functions. So far, practical IND-CCA public-key encryption schemes have been based on specific hard problems associated with specific (algebraic) structures such as hash proof systems.

1.3.4. *Hybrid Usage of Asymmetric and Symmetric Encryptions*

Asymmetric encryption schemes are usually used only for secretly transmitting a session key of a symmetric encryption scheme for message encryption. In fact, the hybrid usage of asymmetric and symmetric encryption schemes is very common in practice. However, this subject had not been well studied, when [18] was published. Only a few papers addressed this topic, for instance that by Abdalla, Bellare, and Rogaway [2], and to the best of our knowledge, no generic construction of secure hybrid usage of asymmetric and symmetric encryption schemes had been proposed.

Abdalla et al. presented a hybrid encryption scheme based on the Diffie–Hellman key-distribution system, first called DHAES [2] and later referred to as DHIES [1].

Shoup presented a framework for generic construction of hybrid encryption schemes, called the KEM/DEM framework [45]. A notable difference from ours is that it enables the sender to create the encrypted session key independent of messages (called the on-the-fly property). However, symmetric encryption schemes, called the data encryption mechanism (DEM) in [45], is required to meet a stronger condition than ours. Abe, Gennaro, and Kurosawa [3] proposed another framework for hybrid usages, called the Tag-KEM/DEM framework. Tag-KEM/DEM does not support the on-the-fly property, and the security requirement of DEM is the same as that of our starting symmetric encryption schemes.

1.4. *Refinement*

In this paper we slightly modify the conference version [18] by replacing "m" with "c" in the second argument of hash H, i.e., $H(\sigma, m)$ with $H(\sigma, c)$. In [18] the starting symmetric encryption scheme should be *deterministic* and *bijective*, but such restrictions have been removed in this full version. We note that it is not written in [18] that the starting symmetric encryption scheme should be limited to bijection, but it is an obvious bug. A counter example is as follows: Let $\Pi = (\mathcal{E}, \mathcal{D})$ be a OT-secure deterministic symmetric encryption scheme derived from a random permutation over $\{0,1\}^k$. Then define $\mathcal{E}'_a(x) := \mathcal{E}_a(x)$ for $x \in \{0,1\}^k$ and $\mathcal{D}'_a(y) := \mathcal{D}_a([y]^k)$ for $y \in \{0,1\}^*$, where $[y]^k$ denotes the first k-bit string of y. $\Pi' = (\mathcal{E}', \mathcal{D}')$ is still deterministic and OT-secure, but the resulting hybrid encryption scheme is not IND-CCA. For completeness, we give the formal proof to the conference version [18] in Appendix B, if starting with a bijective symmetric encryption scheme. We note that the hybrid encryption scheme obtained in [18] is PA2, whereas the hybrid encryption scheme obtained in this full version does not meet even PA1.

2. Preliminary

Let $\{0,1\}^k$ $(k \in N)$ be the set of all the k-bit strings. Let $\{0,1\}^*$ be the set of all finite strings. Conventionally, we include the empty string (denoted ε) in $\{0,1\}^*$. For $x \in \{0,1\}^*$, $|x|$ denotes the bit length of string x. In particular, $|\varepsilon| = 0$.

We write $x := a$ to denote the operation of assigning the value of a to the variable x. Let X be a probability space on finite set $\mathcal{S}(\subset \{0,1\}^*)$. We denote by $x \leftarrow X$ as the operation of sampling an element of $\mathcal{S}$ according to the distribution of X,

and assigning the result of this experiment to the variable x. We also write, for a finite set $\mathcal{S}$, $x \leftarrow_R \mathcal{S}$ to denote the operation of sampling an element of $\mathcal{S}$ uniformly, and assigning the result of this experiment to the variable x. For probability spaces, $X_1, \ldots, X_k$, and k-ary predicate ϕ, we write $\Pr[x_1 \leftarrow X_1; x_2 \leftarrow X_2; \cdots : \phi(x_1, \ldots, x_k)]$ to denote the probability that predicate $\phi(x_1, \ldots, x_k)$ is true after the experiments, "$x_1 \leftarrow X_1; x_2 \leftarrow X_2; \cdots$", are executed in that order. In this case, it is important that $x_1, \ldots, x_k$ are sampled in that order. For probability space X, we write $\Pr[X = a]$ to denote the probability that $\Pr[x \leftarrow X : x = a]$. For probability spaces, $X_1, \ldots, X_k$, and X, we write $\Pr[x_1 \leftarrow X_1; x_2 \leftarrow X_2; \cdots : X(x_1, \ldots, x_k) = a]$ to denote $\Pr[x_1 \leftarrow X_1; x_2 \leftarrow X_2; \cdots; x \leftarrow X : x = a]$.

Let A be a probabilistic algorithm. We write $y \leftarrow A(x_1, \ldots, x_n)$ to denote the experiment of running A for given $(x_1, \ldots, x_n)$, picking r uniformly from an appropriate domain, and assigning the result of this experiment to the variable y, i.e., $y = A(x_1, \ldots, x_n; r)$. Hence, for given fixed input $(x_1, \ldots, x_n)$, we may think of $A(x_1, \ldots, x_n)$ as a probability space. The running time of A denotes the worst-case running time in which algorithm A halts for the same length of input. Algorithm A being t-time means that its running time is t.

Let $\epsilon, \tau : \mathbb{N} \quad [0, 1]$ ($\subset \mathbb{R}$) be positive $[0, 1]$-valued functions. We say that $\epsilon(k)$ is negligible in k if, for any constant c, there exists a constant, $k_0 \in \mathbb{N}$, such that $\epsilon(k) < (1/k)^c$ for any $k > k_0$. Let $f, g : \mathbb{N} \quad \mathbb{R}$. We say that $f = O(g)$ if $\lim_{k \quad \infty} f(k)/g(k) = c$ for some fixed constant c, and that $f = \quad (g)$ if $\lim_{k \quad \infty} f(k)/g(k) = \infty$. We note that $k^{-f(k)}$ is negligible if $f = \quad (1)$.

3. Syntax of Encryption Schemes

We recall the syntax of asymmetric and symmetric encryption schemes, basically following [7,24].

3.1. *Asymmetric Encryption*

An asymmetric (aka public-key) encryption scheme is given by a triple of algorithms, $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, where for every sufficiently large $k \in \mathbb{N}$,

- $\mathcal{K}$, the key-generation algorithm, is a probabilistic polynomial-time (in k) algorithm which on input 1^k outputs a pair of strings, (pk, sk), called the public and secret keys, respectively. This experiment is written as $(pk, sk) \leftarrow \mathcal{K}(1^k)$.
- $\mathcal{E}$, the encryption algorithm, is a probabilistic polynomial-time (in k) algorithm that takes public key pk and message $x \in \mathtt{MSP}$, draws coins r uniformly from coin space $\mathtt{COIN}$, and produces ciphertext $y := \mathcal{E}_{pk}(x; r)$. This experiment is written as $y \leftarrow \mathcal{E}_{pk}(x)$. The message and coin spaces, $\mathtt{MSP}$ and $\mathtt{COIN}$, are uniquely determined by pk.
- $\mathcal{D}$, the decryption algorithm, is a deterministic polynomial-time (in k) algorithm that takes secret key sk and ciphertext $y \in \{0, 1\}^*$, and returns message $x := \mathcal{D}_{sk}(y)$.

We require that an asymmetric encryption scheme should satisfy the following *correctness* condition: For every sufficiently large $k \in \mathbb{N}$, every (pk, sk) generated by $\mathcal{K}(1^k)$ and every $x \in \mathtt{MSP}$, we always have $\mathcal{D}_{sk}(\mathcal{E}_{pk}(x)) = x$.

3.2. Symmetric Encryption

A symmetric (aka private-key) encryption scheme is given by a pair of algorithms, $\Pi = (\mathcal{E}, \mathcal{D})$, where for every sufficiently large $k \in \mathbb{N}$,

- $\mathcal{E}$, the encryption algorithm, is a probabilistic polynomial-time (in k) algorithm that takes secret key $a \in \mathtt{KSP}$ and message $x \in \mathtt{MSP}$, draws coins r uniformly from coin space $\mathtt{COIN}$, and produces ciphertext $y := \mathcal{E}_a(x; r)$. This experiment is written as $y \leftarrow \mathcal{E}_a(x)$. The key, message, and coin spaces, $\mathtt{KSP}$, $\mathtt{MSP}$ and $\mathtt{COIN}$, are uniquely determined by k.
- $\mathcal{D}$, the decryption algorithm, is a deterministic polynomial-time (in k) algorithm that takes secret key $a \in \mathtt{KSP}$ and ciphertext $y \in \{0,1\}^*$, and outputs message $x := \mathcal{D}_a(y)$.

We require that a symmetric encryption scheme should satisfy the *correctness* condition: For every sufficiently large $k \in \mathbb{N}$, every $a \in \mathtt{KSP}$ and every $x \in \mathtt{MSP}$, we always have $\mathcal{D}_a(\mathcal{E}_a(x)) = x$.

In the preliminary version [18], the symmetric encryption schemes utilized for the conversion should be deterministic and bijective, but the restriction is now removed.

4. Generic Conversion

Let $\Pi^{\mathsf{asy}} = (\mathcal{K}^{\mathsf{asy}}, \mathcal{E}^{\mathsf{asy}}, \mathcal{D}^{\mathsf{asy}})$ and $\Pi^{\mathsf{sy}} = (\mathcal{E}^{\mathsf{sy}}, \mathcal{D}^{\mathsf{sy}})$ be asymmetric and symmetric encryption schemes, respectively, (pk, sk) be a pair of public and secret keys generated by $\mathcal{K}^{\mathsf{asy}}(1^k)$, $\mathtt{MSP}^{\mathsf{asy}}$ and $\mathtt{COIN}^{\mathsf{asy}}$ be the message and coin spaces of Π^{asy} with respect to pk, and $\mathtt{KSP}^{\mathsf{sy}}$ and $\mathtt{MSP}^{\mathsf{sy}}$ be the key and message spaces of Π^{sy} (with respect k). We define two hash functions.

$$G : \{0,1\}^* \quad \mathtt{KSP}^{\mathsf{sy}} \quad \text{and} \quad H : \{0,1\}^* \times \{0,1\}^* \quad \mathtt{COIN}^{\mathsf{asy}}.$$

Given Π^{asy} and Π^{sy}, we construct a hybrid encryption scheme $\Pi^{\mathsf{hy}} = (\mathcal{K}^{\mathsf{hy}}, \mathcal{E}^{\mathsf{hy}}, \mathcal{D}^{\mathsf{hy}})$. We write $\mathtt{COIN}^{\mathsf{hy}}$ and $\mathtt{MSP}^{\mathsf{hy}}$ to denote the coin and message spaces of Π^{hy}. This hybrid encryption scheme is specified as follows:

Key-generation $\mathcal{K}^{\mathsf{hy}}$, the key-generation algorithm, takes 1^k as input. It selects $(pk, sk) \leftarrow \mathcal{K}^{\mathsf{asy}}(1^k)$ and returns (pk, sk) as the output of $\mathcal{K}^{\mathsf{hy}}(1^k)$. We write the experiment as $(pk, sk) \leftarrow \mathcal{K}^{\mathsf{hy}}(1^k)$.

Encryption $\mathcal{E}^{\mathsf{hy}}$, the encryption algorithm, takes public key pk and message $m \in \mathtt{MSP}^{\mathsf{hy}}(:= \mathtt{MSP}^{\mathsf{sy}})$ as input. It selects $\sigma \leftarrow_R \mathtt{COIN}^{\mathsf{hy}}(:= \mathtt{MSP}^{\mathsf{asy}})$, computes $c \leftarrow \mathcal{E}^{\mathsf{sy}}_a(m)$, where $a := G(\sigma)$, and computes $e := \mathcal{E}^{\mathsf{asy}}_{pk}(\sigma; h)$ where $h := H(\sigma, c)$. It finally outputs $e \parallel c$ as $\mathcal{E}^{\mathsf{hy}}_{pk}(m; \sigma)$. As described above, the coin and message spaces of Π^{hy} with respect to pk are defined as $\mathtt{COIN}^{\mathsf{hy}} := \mathtt{MSP}^{\mathsf{asy}}$ and $\mathtt{MSP}^{\mathsf{hy}} := \mathtt{MSP}^{\mathsf{sy}}$.

Decryption $\mathcal{D}^{\mathsf{hy}}$, the decryption algorithm, takes secret key sk and ciphertext $e \parallel c \in \{0,1\}^*$ as input. It runs as follows.

1. Parse $e \parallel c$ appropriately as (e, c); otherwise, output ε and halt.
2. Compute $\hat{\sigma} := \mathcal{D}^{\mathsf{asy}}_{sk}(e)$.
3. If $\hat{\sigma} \in \mathtt{COIN}^{\mathsf{hy}}$,
   (a) then compute $\hat{a} := G(\hat{\sigma})$.
   (b) otherwise, set $\mathcal{D}^{\mathsf{hy}}_{sk}(e \parallel c) := \varepsilon$ and go to Step 6.
4. Set $\hat{h} := H(\hat{\sigma}, c)$.
5. If $e = \mathcal{E}^{\mathsf{asy}}_{pk}(\hat{\sigma}; \hat{h})$,
   (a) then set $\mathcal{D}^{\mathsf{hy}}_{sk}(e \parallel c) := \mathcal{D}^{\mathsf{sy}}_{\hat{a}}(c)$.
   (b) otherwise, $\mathcal{D}^{\mathsf{hy}}_{sk}(e \parallel c) := \varepsilon$.
6. Return $\mathcal{D}^{\mathsf{hy}}_{sk}(e \parallel c)$.

4.1. *Remarks on Decryption*

We stress that the error symbol (e.g., ε) used in Step 3 must be the same as that in Step 5; otherwise, the decryption algorithm might reveal crucial information—actually, when the conversion is applied to Okamoto–Uchiyama encryption [35], called EPOC [36], it is crucial, which is reported by Joye, Quisquater, and Yung [27]. Based on the technique [27], Sakurai and Takagi [43] proposed a side-channel attack on EPOC. The essence of the attack is to detect in which of the two steps invalid ciphertexts are rejected, by observing the difference between their computational times. Galindo et al. [21] presented a remedy that modifies the decryption algorithm of EPOC so that it always executes Step 5 and spends almost the same computational time for decryption.

We note that the error symbol in Step 1 can be an arbitrary public symbol. In addition, the error symbol of $\mathcal{D}^{\mathsf{sy}}$ can be arbitrary (but we remark the error symbol of $\mathcal{D}^{\mathsf{sy}}$ can depend only on its inputs and its own description given beforehand).

5. Security Definitions

In this section we define several security notions for asymmetric and symmetric encryption schemes.

5.1. *One-way Asymmetric Encryption*

We give a weak security notion for an asymmetric encryption. Let Π be an asymmetric encryption scheme. We consider the following game of Π against adversary A: Run $\mathcal{K}$ on input 1^k and obtain $(pk, sk) \leftarrow \mathcal{K}(1^k)$. Then pick up $x \leftarrow_R \mathtt{MSP}$ uniformly to compute $y \leftarrow \mathcal{E}_{pk}(x)$. Run adversary A on input (pk, y) to output a string as the decryption of y with pk. The advantage of A is denoted by the probability that A succeeds in decrypting a given encryption of a random plaintext. A is assumed to be *passive*, i.e., A is not allowed to access the decryption oracle.

Definition 5.1 (OWE). Let Π be an asymmetric encryption scheme. Let A be a t-time adversary. For $k \in \mathbb{N}$, define the advantage of A as

$$\mathsf{Adv}^{\mathsf{owe}}_{A,\Pi}(k) \triangleq \Pr\left[(pk, sk) \leftarrow \mathcal{K}\left(1^k\right); x \leftarrow \mathtt{MSP}; y \leftarrow \mathcal{E}_{pk}(x) : A(pk, y) = \mathcal{D}_{sk}(y)\right].$$

We say that Π is (t, ϵ)-OWE secure if, for every t-time adversary A, $\mathsf{Adv}^{\mathsf{owe}}_{A,\Pi}(k) < \epsilon$. In particular, if t is bounded by some polynomial in k and ϵ is negligible in k, we say that Π is one-way.

5.2. *Well-Spread Encryption*

Let $\|X\|$ be the infinity norm of probability space X on a finite set S, i.e., $\|X\| = \max_{a \in \mathcal{S}}\{\Pr[x \leftarrow X : x = a]\}$. The min-entropy of X is $-\log \|X\|$.

Definition 5.2 (γ-spread). Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an asymmetric encryption scheme. For pk and $x \in \mathtt{MSP}$, define the min-entropy of $\mathcal{E}_{pk}(x)$ by $\gamma(pk, x) = -\log \|\mathcal{E}_{pk}(x)\|$, where

$$\left\|\mathcal{E}_{pk}(x)\right\| = \max_{y \in \{0,1\}^*} \Pr\left[h \leftarrow_R \mathtt{COIN} : y = \mathcal{E}_{pk}(x, h)\right].$$

We say that Π is γ-spread (for $k \in \mathbb{N}$), if, for every pk generated by $\mathcal{K}(1^k)$ and $x \in \mathtt{MSP}$, $\gamma(pk, x) \geq \gamma$. In particular, we say that Π is well-spread in k if $\gamma = \quad (\log(k))$.

5.3. *Random Oracle*

We only treat random oracles mapping its inputs to bit strings of a fixed length. For a random oracle, $H : \{0,1\}^* \quad \{0,1\}^n$, we write $H \leftarrow \Omega$ to denote the following imaginary experiment: For each finite string x as a query, sample a uniformly random k-bit string, and assign it to each variable $H(x)$. Without loss of generality, a single random oracle can be treated as multiple, mutually independent random oracles, by appending a fixed distinct bit string to the beginning of each x, i.e., $H_i(x) := H(i\|x)$.

We often require random oracles mapping to a specific finite space, say K, which may depend on the result of executing a specific algorithm. Formally in the random oracle model, the random oracle is chosen *beforehand*. We implicitly assume that we only treat K such that we can appropriately convert a true random oracle into a *pseudo* random oracle mapping to K. For example, let K be a finite cyclic group with order q and generator g. We define $H' : \{0,1\}^* \quad K$ as $H'(x) := g^{H(x) \bmod q}$. Then, the output distribution of H' is statistically close to that of a random oracle mapping to K if $n = \log K + \quad (\log k)$.

For simplicity, we allow ourselves to see such a pseudo random oracle as a true random oracle.

5.4. *Chosen-Ciphertext Security (IND-CCA)*

We recall the chosen-ciphertext security, denoted as IND-CCA or IND-CCA2, for asymmetric encryption [7,40]. In this security notion, we consider a game of asymmetric encryption scheme Π against adversary $A = (A_1, A_2)$ as follows: A_1 takes public key pk and queries decryption oracle $\mathcal{D}_{sk}(\cdot)$. A_1 finally returns two distinct messages, m_0, m_1, as well as some state information s. We pick up random bit $b \in \{0,1\}$ and compute the encryption of m_b, denoted as $c^* = \mathcal{E}_{pk}(m_b)$. Then A_2 takes as input c^* and the above state information s. A_2 can query decryption oracle $\mathcal{D}_{sk}(\cdot)$ with the only restriction that it cannot query the oracle on the challenge ciphertext c^*, and finally guesses b.

The advantage of A is meant by how well it can determine value b by a probability of more than $\frac{1}{2}$. In the random oracle version of IND-CCA, A is allowed to access random oracles in the course of the attack.

Definition 5.3 (IND-CCA in RO). Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an asymmetric encryption scheme, and $A = (A_1, A_2)$ be an adversary for Π. For $k \in \mathbb{N}$, define the following advantage:

$$\mathsf{Adv}^{\text{ind-cca}}_{A,\Pi}(k) = 2 \cdot \Pr\big[H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}\big(1^k\big); (m_0, m_1, s) \leftarrow A_1^{H,\mathcal{D}_{sk}}(pk); \\ b \leftarrow_R \{0,1\}; c^* \leftarrow \mathcal{E}_{pk}(m_b) : A_2^{H,\mathcal{D}_{sk}}\big(c^*, s\big) = b\big] - 1.$$

We say that Π is $(t, q_{\mathtt{hash}}, q_{\mathtt{dec}}, \epsilon)$-IND-CCA secure in the random oracle model if, for every $(t, q_{\mathtt{hash}}, q_{\mathtt{dec}})$-adversary A, $\mathsf{Adv}^{\text{ind-cca}}_{A,\Pi}(k) < \epsilon$, where A is called a $(t, q_{\mathtt{hash}}, q_{\mathtt{dec}})$-adversary if it is a t-time adversary that accesses the random oracle at most $q_{\mathtt{hash}}$ times and the decryption oracle at most $q_{\mathtt{dec}}$ times. In particular, we say that Π is chosen-ciphertext secure in the random oracle model if $(t, q_{\mathtt{hash}}, q_{\mathtt{dec}})$ are bounded by some polynomial in k and ϵ is negligible in k.

5.5. *One-Time Secure Symmetric Encryption*

We introduce a weak security notion for symmetric encryption, called one-time security, which is the symmetric encryption version of indistinguishability against passive attacks. In the preliminary version [18], it is referred to as find–guess security.

Let $\Pi = (\mathcal{E}, \mathcal{D})$ be a symmetric-key encryption scheme, and $A = (A_1, A_2)$ be an adversary against Π. We consider a game of Π against adversary A as follows:

1. Pick up key $a \leftarrow \mathtt{KSP}$.
2. Run A_1 on input 1^k. Finally, A_1 outputs two distinct messages, $m_0, m_1 \in \mathtt{MSP}$ with some state information s.
3. Pick up random bit $b \leftarrow_R \{0,1\}$ and compute $c \leftarrow \mathcal{E}_a(m_b)$.
4. Run A_2 on input c and s. A_2 finally outputs $b' \in \{0,1\}$.

The advantage of A indicates how much better it can determine the value b by a probability of more than $\frac{1}{2}$, namely $2\Pr[b = b'] - 1$. A is just passive, i.e., not allowed to access any encryption or decryption oracle.

Definition 5.4 (OT). Let $\Pi = (\mathcal{E}, \mathcal{D})$ be a symmetric-key encryption scheme and let A be an adversary that works on Π. For $k \in \mathbb{N}$, define the advantage of A, by

$$\mathsf{Adv}^{\mathsf{ot}}_{A,\Pi}(k) = 2 \cdot \Pr\big[a \leftarrow_R \mathtt{KSP}(k); (m_0, m_1, s) \leftarrow A_1\big(1^k\big); b \leftarrow_R \{0,1\}; \\ c \leftarrow \mathcal{E}_a(m_b) : A_2(c, s) = b\big] - 1.$$

We say that Π is (t, ϵ)-OT secure if, for every t-time adversary A, $\mathsf{Adv}^{\mathsf{ot}}_{A,\Pi}(k) < \epsilon$. In particular, if t is bounded by some polynomial in k and ϵ is negligible in k, we say that Π is one-time secure.

6. Security Results

Let Π^{asy} be a γ-spread $(t^{\mathsf{asy}}, \epsilon^{\mathsf{asy}})$-OWE secure asymmetric encryption scheme. Let Π^{sy} be a $(t^{\mathsf{sy}}, \epsilon^{\mathsf{sy}})$-OT secure symmetric encryption scheme. $T^{\mathsf{asy}}(k)$ denotes the worst case of the running time of $\mathcal{E}^{\mathsf{asy}}_{pk}$ for every pk generated by $\mathcal{K}^{\mathsf{asy}}(1^k)$. Let Π^{hy} be the hybrid encryption scheme obtained by our conversion. We then have the following theorem.

Theorem 6.1. *Π^{hy} is $(t^{\mathsf{hy}}, q_{\mathtt{hash}}, q_{\mathtt{dec}}, \epsilon^{\mathsf{hy}})$-IND-CCA secure in the random oracle model where*

$$t^{\mathsf{hy}} = \min\left(t^{\mathsf{asy}}, t^{\mathsf{sy}}\right) - (q_{\mathtt{dec}} + 1)T^{\mathsf{asy}}(k) - q_{\mathtt{hash}}O(k) \quad \textit{and}$$
$$\epsilon^{\mathsf{hy}} = 2q_{\mathtt{hash}}\epsilon^{\mathsf{asy}} + \epsilon^{\mathsf{sy}} + 2q_{\mathtt{dec}}2^{-\gamma}.$$

The proof is given in Sect. 7.

Suppose that Π^{sy} is a *one-time pad* based symmetric encryption scheme defined over $\{0,1\}^k$. Namely,

- $\mathtt{KSP}^{\mathsf{sy}} = \mathtt{MSP}^{\mathsf{sy}} = \{0,1\}^k$.
- For $m \in \{0,1\}^k$, $\mathcal{E}^{\mathsf{sy}}_a(m) := a \oplus m$.
- For $c \in \{0,1\}^k$, $\mathcal{D}^{\mathsf{sy}}_a(c)$ outputs $a \oplus c$.

We note that for $m \notin \{0,1\}^k$ or $c \notin \{0,1\}^k$, the encryption and decryption algorithms can be specified in an arbitrary way, as long as Π^{sy} is secure in the sense of OT.

Corollary 6.2. *Let the underlying symmetric encryption scheme be "one-time padding" defined above. Let $G : \{0,1\}^* \quad \{0,1\}^k$ and $H : \{0,1\}^* \times \{0,1\}^* \quad \{0,1\}^k$. The hybrid encryption scheme is $(t^{\mathsf{hy}}, q_{\mathtt{hash}}, q_{\mathtt{dec}}, \epsilon^{\mathsf{hy}})$- IND-CCA secure in the random oracle model where*

$$t^{\mathsf{hy}} = t^{\mathsf{asy}} - (q_{\mathtt{dec}} + 1)T^{\mathsf{asy}}(k) - q_{\mathtt{hash}}O(k) \quad \textit{and}$$
$$\epsilon^{\mathsf{hy}} = 2q_{\mathtt{hash}}\epsilon^{\mathsf{asy}} + 2q_{\mathtt{dec}}2^{-\gamma}.$$

Proof. The above one-time pad based symmetric encryption scheme is $(\infty, 0)$-OT secure. It straightforwardly implies the corollary. □

Note that any γ-spread asymmetric encryption scheme Π^{asy} can be transformed to $(\gamma + \gamma')$-spread asymmetric encryption scheme $\Pi^{\hat{\mathsf{asy}}}$ for any γ' by appending random $r' \in \{0,1\}^{\gamma'}$ to the end of the encryption of a message:

$$\hat{\mathcal{E}}^{\mathsf{asy}}_{pk}\left(m; \left(r \| r'\right)\right) = \mathcal{E}^{\mathsf{asy}}_{pk}(m; r) \parallel r'.$$

The new asymmetric encryption scheme is OWE if so is the original one. Thus, we have the following statement.

Corollary 6.3. *There exists an efficient transformation from any one-way asymmetric encryption into an asymmetric encryption scheme that is secure in the sense of IND-CCA in the random oracle model.*

7. Proof of Theorem 6.1

7.1. *Outline of Proof*

Let A^{hy} be a $(t^{\mathsf{hy}}, q_{\mathtt{hash}}, q_{\mathtt{dec}})$-adversary with advantage ϵ^{hy} that attacks Π^{hy} in the sense of IND-CCA. We refer to the attack game as Game G_0.

We simply adopt the so-called game-hopping or game-playing technique used by Cramer and Shoup [13]. (Prior the work of [13], a similar framework is used in [28] and these techniques have the origin in the earliest hybrid argument.) We define a sequence of modified attack games, $G_1, \ldots, G_l$. Each game operates on the same probability space determined by the following mutually independent random variables:

1. the choices of random oracles, $G, H \in \mathtt{Hash}$;
2. the contents of random tapes of $A^{\mathsf{hy}} = (A_1^{\mathsf{hy}}, A_2^{\mathsf{hy}})$, denoted $r^{A^{\mathsf{hy}}}$;
3. the random variables in the environment: $(pk, sk) \in \mathcal{K}^{\mathsf{asy}}(1^k)$, $b \in \{0,1\}$, $\sigma^* \in \mathtt{MSP}^{\mathsf{asy}}$ and $r^* \in \mathtt{COIN}^{\mathsf{sy}}$; and
4. the following *imaginary* random variables: $a^* \in \mathtt{KSP}^{\mathsf{sy}}$ and $h^* \in \mathtt{COIN}^{\mathsf{asy}}$.

Among the above games, the probability spaces are identical. Only some of the rules defining how the environment responds to oracle queries differ from game to game. We write (e^*, c^*) to denote the challenge ciphertext and b' to denote the final output of the adversary (i.e., the output of A_2^{hy}) in the above games. In Game G_0, the challenge ciphertext (e^*, c^*) is generated by computing $c^* := \mathcal{E}^{\mathsf{sy}}_{G(\sigma^*)}(m_b; r^*)$ and $e^* := \mathcal{E}^{\mathsf{asy}}_{pk}(\sigma^*; H(\sigma^*, c^*))$, where (b, σ^*, r^*) are the above random variables.

For any $0 \le i \le l$, we write S_i to denote the event that $b = b'$ in game G_i. Since the rules of the environment responses are generally different among the above games, Event S_i's are different in general. By definition, $\epsilon^{\mathsf{hy}} = 2\Pr[S_0] - 1$. The final goal is to evaluate ϵ^{hy} by some upper-bound. Our strategy for the proof is that we will change Event S_i step by step, by clarifying the upper-bound of $\Pr[S_{i-1}] - \Pr[S_i]$. When we have the exact value of $\Pr[S_l]$ for some l, we achieve the final goal to bound ϵ^{hy}.

For any $0 \le i \le l$, we write $\mathsf{Ask}\sigma_i^*$ to denote the event in Game G_i such that

- A^{hy} submits to random oracle G the query σ^* or
- A^{hy} submits to random oracle H a query with the form of (σ^*, c^*).

Here is the road map of the games and how the proof is going on.

1. Game G_1 is the same game as Game G_0, except that we change the rule of the decryption oracle. We simulate the decryption oracle without using secret key sk. This simulation is described later. Then, we see that $\Pr[S_0] - \Pr[S_1] \le q_{\mathtt{dec}} 2^{-\gamma}$.
2. Game G_2 is the same game as Game G_1, except that we replace the values of $G(\sigma^*)$ and $H(\sigma^*, c^*)$ with a^* and h^*, respectively. This change is only conceptual. So, $\Pr[S_1] - \Pr[S_2] = 0$.
3. Game G_3 is the same game as Game G_2, except that we replace a^* and h^* with $G(\sigma^*)$ and $H(\sigma^*, c^*)$, again, whereas we use the same challenge ciphertext (e^*, c^*) as in Game G_2, i.e., $e^* = \mathcal{E}^{\mathsf{asy}}_{pk}(\sigma^*; h^*)$ and $c^* = \mathcal{E}^{\mathsf{sy}}_{a^*}(m_b)$. Then, we have $\mathsf{Ask}\sigma_2^* = \mathsf{Ask}\sigma_3^*$ and $\Pr[S_2] - \Pr[S_3] \le \Pr[\mathsf{Ask}\sigma_3^*]$.

4. Game G_4 is the same as Game G_3, except that we change the rule of selecting random oracles. This change is only conceptual. Thus, we have $S_3 = S_4$ and $\mathsf{Ask}\,\sigma_3^* = \mathsf{Ask}\,\sigma_4^*$.

Then, we see that $\Pr[\mathsf{Ask}\,\sigma_4^*] \leq q_{\mathtt{hash}}\epsilon^{\mathsf{asy}}$, if t^{asy} is enough large. We also see that $\Pr[S_4] = \Pr[\mathsf{Succ}\,A^{\mathsf{sy}}]$, if t^{sy} is enough large, where we let $\mathsf{Succ}\,A^{\mathsf{sy}}$ be the event in game G_{3s} that

$$a \leftarrow_R \mathtt{KSP}(k); \qquad (m_0, m_1, s) \leftarrow A_1^{\mathsf{sy}}\big(1^k\big); \qquad b \leftarrow_R \{0,1\};$$
$$c \;\leftarrow\; \mathcal{E}_a^{\mathsf{sy}}(m_b) : A_2^{\mathsf{sy}}(c, s) = b.$$

Tracing through the above steps, we can see that $\Pr[S_0] \leq \Pr[\mathsf{Succ}\,A^{\mathsf{sy}}] + q_{\mathtt{hash}}\epsilon^{\mathsf{asy}} + q_{\mathtt{dec}}2^{-\gamma}$. Hence,

$$\epsilon^{\mathsf{hy}} \leq 2q_{\mathtt{hash}}\epsilon^{\mathsf{asy}} + \epsilon^{\mathsf{sy}} + 2q_{\mathtt{dec}}2^{-\gamma}, \tag{1}$$

where $\epsilon^{\mathsf{sy}} = 2\Pr[\mathsf{Succ}\,A^{\mathsf{sy}}] - 1$.

Before proceeding the detail, we prepare the following lemma.

Lemma 7.1 [13]. *Let A, B, F_1, F_2 be events defined on the same probability space. Suppose that $\Pr[F_1] = \Pr[F_2]$ and $\Pr[A \wedge \neg F_1] = \Pr[B \wedge \neg F_2]$. Then we have $\Pr[A] - \Pr[B] \leq \Pr[F_1] (= \Pr[F_2])$.*

Proof. By the condition we have $\Pr[A] - \Pr[B] = \Pr[A \wedge F_1] - \Pr[B \wedge F_2]$. Since $\Pr[A \wedge F_1] \leq \Pr[F_1]$ and $\Pr[B \wedge F_2] \geq 0$, we have $\Pr[A] - \Pr[B] \leq \Pr[F_1]$. □

7.2. *Details of Reduction*

We now provide details.

Game G_1 Game G_1 is identical to game G_0 except for changing the rule of how to reply for decryption queries. Let $\mathcal{G}$ and $\mathcal{H}$ be the query/answer record lists for random oracles, G and H, respectively.

For fresh query (e, c) to the decryption oracle, we proceed as follows:

1. Search tuple (σ, c, h) in $\mathcal{H}$ such that $\sigma \in \mathtt{MSP}^{\mathsf{asy}}$ and $e = \mathcal{E}_{pk}^{\mathsf{asy}}(\sigma; h)$. If such a tuple is not recorded in $\mathcal{H}$, return ε; otherwise.
2. Query oracle G on σ to obtain $a = G(\sigma)$.
3. Return $\mathcal{D}_a^{\mathsf{sy}}(c)$.

If query (e, c) is not new, we just return the same value as before. Note that this procedure does not require the secret key.

We say that (e, c) is *almost-valid* with respect to pk on Π^{hy}, if and only if we have

$$e = \mathcal{E}_{pk}^{\mathsf{asy}}\big(\sigma; H(\sigma, c)\big), \quad \text{where } \sigma := \mathcal{D}_{sk}^{\mathsf{asy}}(e). \tag{2}$$

We note that an almost-valid ciphertext (e, c) is *valid* (w.r.t. pk on Π^{hy}) if $\sigma \in \mathtt{MSP}^{\mathsf{asy}}$.

Consider the case that "A^{hy} submits almost-valid ciphertext (e, c) to the decryption oracle, whereas there is no (σ, c, h) in $\mathcal{H}$ such that $\sigma \in \mathtt{MSP}^{\mathsf{asy}}$ and $e = \mathcal{E}_{pk}^{\mathsf{asy}}(\sigma; h)$." Let

Bad_0 and Bad_1 be the events that the case occurs in games, G_0 and G_1, respectively. Conditioned on Event Bad_1, the ciphertext is always rejected in Game G_1, whereas conditioned on Event Bad_0, it is not rejected in Game G_0 if the almost-valid ciphertext is valid.

By construction, games, G_0 and G_1, proceed identically until the above case occurs. This implies that $\mathsf{Bad}_0 = \mathsf{Bad}_1$ and $S_0 \wedge \neg\mathsf{Bad}_0 = S_1 \wedge \neg\mathsf{Bad}_1$. Hence, by Lemma 7.1, we have $\Pr[S_0] - \Pr[S_1] \leq \Pr[\mathsf{Bad}_0]$.

We consider the probability that Event Bad_0 occurs. Remember that (e^*, c^*) denotes the challenge ciphertext. Since challenge ciphertext (e^*, c^*) is valid, we always have $\sigma^* = \mathcal{D}^{\mathsf{asy}}_{sk}(e^*)$.

- Suppose that e is a *valid* ciphertext of Π^{asy} with respect to pk, i.e., there exists $\sigma \in \mathtt{MSP}^{\mathsf{asy}}$ and $r \in \mathtt{COIN}^{\mathsf{asy}}$ such that $e = \mathcal{E}^{\mathsf{asy}}_{pk}(\sigma; r)$. Then we always have $\sigma \neq \sigma^*$ or $c \neq c^*$: If $e = e^*$ (which implies $\sigma = \sigma^*$), we must have $c \neq c^*$, because $(e, c) \neq (e^*, c^*)$. If $e \neq e^*$, must have $c \neq c^*$ or $\sigma \neq \sigma^*$; otherwise, $H(\sigma, c) = H(\sigma^*, c^*)$. Thus, we have $e = e^* = \mathcal{E}^{\mathsf{asy}}_{pk}(\sigma; H(\sigma, c))$, which contradicts the condition. Hence, given ciphertext (e, c) with valid e, the value $H(\sigma, c)$ is independent of the value $H(\sigma^*, c^*)$, because $(\sigma, c) \neq (\sigma^*, c^*)$. Therefore, the probability of Bad_0 on this condition is at most $q_{\mathtt{dec}} 2^{-\gamma}$ because Π^{asy} is assumed to be γ-spread.
- On the contrary, if e is an invalid ciphertext of Π^{asy} with respect to pk, we should consider the event that $\sigma = \sigma^*$ and $c = c^*$ both occurs, despite the fact $(e, c) \neq (e^*, c^*)$—we cannot immediately deny the possibility that we have $\mathcal{D}^{\mathsf{asy}}_{sk}(e) = \mathcal{D}^{\mathsf{asy}}_{sk}(e^*)$ for some invalid e, with $e \neq e^*$, because the decryption of asymmetric encryption on an invalid ciphertext is not specified. However, this event never occurs, because if it occurs then $\sigma \in \mathtt{MSP}^{\mathsf{asy}}$ and $H(\sigma, c) \in \mathtt{COIN}^{\mathsf{asy}}$ (because $H(\sigma, c) = H(\sigma^*, c^*)$), which contradicts that e is invalid. So if e is invalid, (2) never holds and Bad_0 never occurs.

Hence, $\Pr[\mathsf{Bad}_0] \leq q_{\mathtt{dec}} 2^{-\gamma}$. Therefore, $\Pr[S_0] - \Pr[S_1] \leq q_{\mathtt{dec}} 2^{-\gamma}$.

Game G_2 Game G_2 is the same as Game G_1 except that we replace values, $G(\sigma^*)$ and $H(\sigma^*, c^*)$, with a^* and h^*, respectively. We create the challenge ciphertext (e^*, c^*) such that $e^* = \mathcal{E}^{\mathsf{asy}}_{pk}(\sigma^*; h^*)$ and $c^* = \mathcal{E}^{\mathsf{sy}}_{a^*}(m_b)$, and reply with a^* and h^* if σ^* and (σ^*, c^*) is submitted to G and H, respectively. This means that after we fix G and H, we replace them with G' and H', respectively, where G' and H' are identical to G and H except for the above queries. However, it is clear that the distribution on (G, H) is identical to that of (G', H'). Hence, $\Pr[S_1] = \Pr[S_2]$.

Game G_3 Game G_3 is the same as Game G_2, except that we replace G' and H' with G and H, again, but still use the same challenge ciphertext (e^*, c^*) such that $e^* = \mathcal{E}^{\mathsf{asy}}_{pk}(\sigma^*; h^*)$ and $c^* = \mathcal{E}^{\mathsf{sy}}_{a^*}(m_b)$. Games, G_2 and G_3, proceed identically, until the adversary asks G for σ^* to obtain $G(\sigma^*)$ or H for (σ^*, c^*) to obtain $H(\sigma^*, c^*)$. Thus, we see that $\mathsf{Ask}\,\sigma_2^* = \mathsf{Ask}\,\sigma_3^*$ and $S_2 \wedge \neg\mathsf{Ask}\,\sigma_2^* = S_3 \wedge \neg\mathsf{Ask}\,\sigma_3^*$. By Lemma 7.1, we have $\Pr[S_2] - \Pr[S_3] \leq \Pr[\mathsf{Ask}\,\sigma_3^*]$.

Game G_4 In Game G_4, we modify the rule of selecting random oracles. Instead of selecting random functions, G and H, in advance, we select their values step by step as follows.

- Let $\mathcal{G}$ be the query/answer list for oracle G. $\mathcal{G}$ is initially empty. For a fresh query σ to G, select $a \in_R \mathtt{KSP}^{\mathsf{sy}}$ to reply with. Add (σ, a) to $\mathcal{G}$.
- Let $\mathcal{H}$ be the query/answer list for oracle H. $\mathcal{H}$ is initially empty. For a fresh query (σ, c) to H, pick up $h \in_R \mathtt{COIN}^{\mathsf{asy}}$ to reply with. Then add the tuple (σ, c, h) to $\mathcal{H}$.

In the above, if the query is not fresh, then simply reply with the same value as that has already been in the list $\mathcal{G}$ or $\mathcal{H}$. It is clear that this change is only conceptual. Therefore, $\Pr[S_3] = \Pr[S_4]$ and $\Pr[\mathsf{Ask}\,\sigma_3^*] = \Pr[\mathsf{Ask}\,\sigma_4^*]$.

We now have $\Pr[S_0] - \Pr[S_4] \le q_{\mathtt{dec}} 2^{-\gamma} + \Pr[\mathsf{Ask}\,\sigma_4^*]$.

Lemma 7.2. *Let* Π^{asy} *be* $(t^{\mathsf{asy}}, \epsilon^{\mathsf{asy}})$-OWE, *where* $t^{\mathsf{asy}} \ge t^{\mathsf{hy}} + q_{\mathtt{dec}} T^{\mathsf{asy}}(k) + q_{\mathtt{hash}} O(k)$. *Then* $\Pr[\mathsf{Ask}\,\sigma_4^*] \le q_{\mathtt{hash}} \epsilon^{\mathsf{asy}}$.

Proof. Suppose that A^{asy} is a t^{asy}-time adversary to attack Π^{asy} in the sense of OWE with advantage ϵ^{asy}, where $t^{\mathsf{asy}} = t^{\mathsf{hy}} + q_{\mathtt{dec}} T^{\mathsf{asy}}(k) + q_{\mathtt{hash}} O(k)$. The advantage of A^{asy} is the probability that "$(pk, sk) \leftarrow \mathcal{K}^{\mathsf{asy}}(1^k); x \leftarrow \mathtt{MSP}^{\mathsf{asy}}; y \leftarrow \mathcal{E}_{pk}^{\mathsf{asy}}(x) : A^{\mathsf{asy}}(pk, y) = \mathcal{D}_{sk}^{\mathsf{asy}}(y)$." We construct A^{asy} by using $A^{\mathsf{hy}} = (A_1^{\mathsf{hy}}, A_2^{\mathsf{hy}})$ in Game G_4 as follows:

1. A^{asy} takes (pk, y) as input, where y is the challenge ciphertext.
2. A^{asy} selects $a^* \leftarrow_R \mathtt{KSP}^{\mathsf{sy}}$ and runs A_1^{hy} on input pk.
3. When A_1^{hy} submits queries to the oracles, G, H and $\mathcal{D}_{sk}^{\mathsf{hy}}$, A^{asy} simulates them as described above in Game G_4. It waits until A_1^{hy} outputs (m_0, m_1, s).
4. A^{asy} selects $b \leftarrow_R \{0, 1\}$ and sets $e^* := y$ and $c^* \leftarrow \mathcal{E}_{a^*}^{\mathsf{sy}}(m_b)$. A^{asy} provides (e^*, c^*, s) for A_2^{hy} and runs A_2^{hy}.
5. When A_2^{hy} submits queries to the oracles, G, H and $\mathcal{D}_{sk}^{\mathsf{hy}}$, A^{asy} simulates them as described above in Game G_4.
6. When A_2^{hy} halts, A^{asy} selects $i \leftarrow_R \{1, \ldots, q_{\mathtt{hash}}\}$ and outputs σ in the ith query to the random oracles.

We note that, conditioned on the event that A^{hy} submits σ^* to G or $(\sigma^*, \cdot)$ to H, A^{asy} outputs σ^* with probability $q_{\mathtt{hash}}^{-1}$.

In order to simulate G and H, A^{asy} should select random values from $\mathtt{KSP}^{\mathsf{sy}}$ and $\mathtt{COIN}^{\mathsf{asy}}$ $q_{\mathtt{hash}}$ times in total, which costs $q_{\mathtt{hash}} O(k)$. Hence, the running time of A^{asy} is $t^{\mathsf{hy}} + q_{\mathtt{dec}} T^{\mathsf{asy}}(k) + q_{\mathtt{hash}} O(k)$. By construction, it is obvious that the view of A^{hy} is identical to that of G_4, if

$$t^{\mathsf{asy}} \ge t^{\mathsf{hy}} + q_{\mathtt{dec}} T^{\mathsf{asy}}(k) + q_{\mathtt{hash}} O(k). \tag{3}$$

Then, thanks to $(t^{\mathsf{asy}}, \epsilon^{\mathsf{asy}})$-OWE Π^{asy}, the probability that A^{asy} outputs σ^* $(:= \mathcal{D}_{sk}^{\mathsf{asy}}(c^*))$ is bounded by ϵ^{asy}. Therefore, we have $\Pr[\mathsf{Ask}\,\sigma_4^*] \le q_{\mathtt{hash}} \epsilon^{\mathsf{asy}}$, because Event $\mathsf{Ask}\,\sigma_4^*$ implies the event that A^{hy} submits σ^* or $(\sigma^*, \cdot)$ to G or H in Game 4. □

Lemma 7.3. *Let* Π^{sy} *be* $(t^{\mathsf{sy}}, \epsilon^{\mathsf{sy}})$-OT, *where* $t^{\mathsf{sy}} \ge t^{\mathsf{hy}} + (q_{\mathtt{dec}} + 1) T^{\mathsf{asy}}(k) + q_{\mathtt{hash}} O(k)$. *Then, we have* $\Pr[S_4] \le \Pr[\mathsf{Succ}\, A^{\mathsf{sy}}]$; *hence*, $2\Pr[S_4] - 1 \le \epsilon^{\mathsf{sy}}$.

Proof. Suppose that A^{sy} is t^{sy}-time adversary to attack Π^{sy} in the sense of OT with advantage ϵ^{sy}, where $t^{\mathsf{sy}} = t^{\mathsf{hy}} + (q_{\mathtt{dec}} + 1)T^{\mathsf{asy}}(k) + q_{\mathtt{hash}}O(k)$. Remember that $\mathsf{Succ}\, A^{\mathsf{sy}}$ is the event that "$a \leftarrow \mathtt{KSP}^{\mathsf{sy}}$; $(m_0, m_1, s) \leftarrow A_1^{\mathsf{sy}}(1^k)$; $b \leftarrow_R \{0,1\}$; $c \leftarrow \mathcal{E}_a^{\mathsf{sy}}(m_b) : A^{\mathsf{sy}}(c, s) = b$." The advantage of A^{sy}, ϵ^{sy}, is $2\Pr[\mathsf{Succ}\, A^{\mathsf{sy}}] - 1$. We construct A^{sy} by using A^{hy} in Game G_4 as follows.

1. A_1^{sy} is given 1^k.
2. A_1^{sy} runs $\mathcal{K}^{\mathsf{asy}}$ on input 1^k to take (pk, sk). A_1^{sy} then runs A_1^{hy} on the public key pk.
3. When A_1^{hy} submits queries to the oracles, G, H and $\mathcal{D}_{sk}^{\mathsf{hy}}$, A_1^{sy} simulates them as described above in Game G_4. It waits until A_1^{hy} outputs (m_0, m_1, s).
4. A_2^{sy} takes the challenge ciphertext $c^* = \mathcal{E}_{a^*}^{\mathsf{sy}}(m_b)$ and state s, where $b \in_R \{0,1\}$.
5. A_2^{sy} selects $\sigma^* \leftarrow_R \mathtt{MSP}^{\mathsf{asy}}$ and $h^* \leftarrow_R \mathtt{COIN}^{\mathsf{asy}}$. It then sets $e^* = \mathcal{E}_{pk}^{\mathsf{asy}}(\sigma^*; h^*)$. A_2^{sy} runs A_2^{hy} on input (e^*, c^*, s).
6. When A_2^{hy} submits queries to the oracles, $G, H, \mathcal{D}_{sk}^{\mathsf{hy}}$, A_2^{sy} simulates them as described above in Game G_4.
7. When A^{hy} outputs bit b' and halts, A_2^{sy} outputs the same b'.

By construction, the view of A^{hy} is identical to that of Game G_4, if

$$t^{\mathsf{sy}} \geq t^{\mathsf{hy}} + (q_{\mathtt{dec}} + 1)T^{\mathsf{asy}}(k) + q_{\mathtt{hash}}O(k). \tag{4}$$

Thus, $\Pr[S_4] = \Pr[\mathsf{Succ}\, A^{\mathsf{sy}}]$. □

To satisfy both (3) and (4), we requires that

$$t^{\mathsf{hy}} \leq \min\left(t^{\mathsf{asy}}, t^{\mathsf{sy}}\right) - (q_{\mathtt{dec}} + 1)T^{\mathsf{asy}}(k) - q_{\mathtt{hash}}O(k).$$

By the lemmas above, when both (3) and (4) hold, $\epsilon^{\mathsf{hy}}, \epsilon^{\mathsf{asy}}, \epsilon^{\mathsf{sy}}$ obey Inequality (1).

Acknowledgements

We would like to thank Phillip Rogaway for his invaluable support in developing the preliminary version of our manuscript. We would also like to thank Yehuda Lindell and Moti Yung for pointing out a few important mistakes in the preliminary version of the paper. We also thank Masayuki Abe for his valuable discussion.

Appendix A. Notes on PA

The notion of plaintext-awareness was first formalized in [6] and later revised in [7] for incompleteness. To distinguish the two definitions, we call the former PA1 and the latter PA2. PA1 implies IND-CCA1 [6], whereas PA2 implies IND-CCA2 [7]. The main property of the notion of plaintext-awareness is informally that the adversary cannot produce a *new* ciphertext without *knowing* the corresponding plaintext. In PA1, the adversary cannot do so *before* it takes the challenge ciphertext, whereas in PA2 it cannot even *after* taking the challenge ciphertext. We say that an asymmetric encryption scheme is PA1

(resp. PA2) if it is IND-CPA, in addition to satisfying the above property. The proofs of the statement that PA1 (resp. PA2) implies IND-CCA1 (resp. IND-CCA2) comes from the following intuitive idea: If the target encryption scheme is PA1 (resp. PA2), an adversary *is aware of* the decryption of the ciphertexts submitted to the decryption oracle. Hence, it cannot obtain any additional information from the decryption oracle because it already knows the corresponding plaintexts. Therefore, we can transform a-priori (resp. a-posteriori) chosen-ciphertext attacks against the target encryption scheme into a chosen plaintext attack against the same encryption scheme. Hence, if the encryption scheme is IND-CPA, it would be IND-CCA1 (resp. IND-CCA2).

The opposite directions do not hold. An artificial counter example appeared in [7], that meets IND-CCA2, but not PA1. Therefore, IND-CCA1 (resp. IND-CCA2) does not imply PA1 (resp. PA2), because PA2 implies PA1 and IND-CCA2 implies IND-CCA1. A more natural counter example was shown by Phan and Pointcheval [38], where OAEP 3-round [38] does not meet PA1, but still remains IND-CCA2. The hybrid encryption scheme obtained in this paper is another natural counter example, which does not meet PA1, but still IND-CCA2 (we note that the hybrid encryption scheme in the conference version [18] is PA2 if the starting symmetric encryption scheme is deterministic and bijective).

Let $f : \{0,1\}^k \quad \{0,1\}^k$ be a trap-door permutation. Consider asymmetric encryption scheme Π induced by the following encryption function: For plaintext $m \in \{0,1\}^k$, to encrypt it as

$$\mathcal{E}_{pk}^{\mathsf{hy}}(m;\sigma) = f(\sigma) \| H(\sigma, c) \| G(\sigma) \oplus m, \tag{A.1}$$

where $G : \{0,1\}^k \quad \{0,1\}^k$ and $c = G(\sigma) \oplus m$. This encryption scheme is a special case of our hybrid encryption scheme. This scheme is IND-CCA2, but not PA1. We let the reader refer to [6,7] for the formal definitions of PA1, but the proof intuitively is as follows: Consider plaintext creator B for Π that takes public-key pk and outputs a ciphertext, after asking queries to the random oracles. It can produce a ciphertext (f, h, c), *without knowing* the corresponding plaintext in the following way—B just picks up any $\sigma, c \in \{0,1\}^k$ and submits (σ, c) to H to obtain $h = H(\sigma, c)$. It then computes $f = f(\sigma)$, using $pk = \{f\}$, and submits, *not asking* G *with* σ, the ciphertext (f, h, c) to the decryption oracle. The decryption for this ciphertext, $G(\sigma) \oplus c$, is unpredictable, because $G(\sigma)$ is unpredictable. Hence, B cannot be aware of the decryption for this ciphertext without accessing the decryption oracle. In other words, one cannot construct a knowledge extractor that outputs the value $\mathcal{D}_{sk}^{\mathsf{hy},G,H}(f, h, c)$ on the transcript $\{(\sigma, c, h), (f, h, c)\}$. If such a knowledge extractor exists for Π^{hy}, it contradicts the unpredictability of $G(\sigma)$. We have the following claim.

Claim A.1. *Let Π be the above encryption scheme. There exists a polynomial-time bounded plaintext creator B so that there is no knowledge extractor for Π such that* $\mathsf{Adv}_{K,B,\Pi}^{\mathsf{ke}}(k) > 2^{-k}$.

Proof. Suppose that K is a knowledge extractor for Π. We construct B as above. We then run K on B's transcript. If K outputs m', then we return $m' \oplus c$ as the value of

$G(\sigma)$. Since G is a random oracle and K is not allowed to access G, the chance of $G(\sigma) = m' \oplus c$ is at most 2^{-k} for any K. Therefore, $\mathsf{Adv}^{\mathsf{ke}}_{K,B,\Pi}(k) \leq 2^{-k}$. □

By this claim, Π does not meet PA1.

Appendix B. Security Proof of [18]

We briefly provide the formal proof of the conversion [18], assuming that the underlying symmetric encryption scheme is deterministic and bijective, in order to complete the statement and show a better security bound than [18]. We say that a symmetric encryption scheme, $\Pi = (\mathcal{E}, \mathcal{D})$, is deterministic if $\mathcal{E}$ is a deterministic algorithm. We say that a deterministic symmetric encryption scheme, $\Pi = (\mathcal{E}, \mathcal{D})$ is bijective if, for any $a \in \mathtt{KSP}$ and any $y \notin \mathcal{E}_a(\mathtt{MSP})$, $\mathcal{D}_a$ rejects y as an invalid ciphertext, where $\mathcal{E}_a(\mathtt{MSP})$ denotes the image of $\mathcal{E}_a$, namely $\mathcal{E}_a(\mathtt{MSP}) \triangleq \{\mathcal{E}_a(x) | x \in \mathtt{MSP}\}$.

Let $\Pi^{\mathsf{asy}} = (\mathcal{K}^{\mathsf{asy}}, \mathcal{E}^{\mathsf{asy}}, \mathcal{D}^{\mathsf{asy}})$ be an asymmetric encryption scheme, and let $\Pi^{\mathsf{sy}} = (\mathcal{E}^{\mathsf{sy}}, \mathcal{D}^{\mathsf{sy}})$ be a deterministic and bijective symmetric encryption scheme. The conference version of the conversion is obtained by replacing "c" with "m" in H in this version. Namely,

$$\mathcal{E}^{\mathsf{hy}}_{pk}(m;\sigma) = \mathcal{E}^{\mathsf{asy}}_{pk}\big(\sigma; H(\sigma, m)\big) \parallel \mathcal{E}^{\mathsf{sy}}_{G(\sigma)}(m).$$

The coin and message spaces of Π^{hy} with respect to pk are defined as $\mathtt{COIN}^{\mathsf{hy}} := \mathtt{MSP}^{\mathsf{asy}}$ and $\mathtt{MSP}^{\mathsf{hy}} := \mathtt{MSP}^{\mathsf{sy}}$. The decryption procedure of $\mathcal{D}^{\mathsf{hy}}_{sk}$ is the same as that for this version except for replacing c with $\hat{m}$ and rejecting the ciphertext if $\hat{m} \notin \mathtt{MSP}^{\mathsf{asy}}$ in Step 5. Namely, $\mathcal{D}^{\mathsf{hy}}_{sk}$ takes ciphertext $e \parallel c \in \{0,1\}^*$ as input and runs as follows.

1. Parse $e \parallel c$ appropriately as (e, c); otherwise, output ε and halt.
2. Compute $\hat{\sigma} := \mathcal{D}^{\mathsf{asy}}_{sk}(c)$.
3. If $\hat{\sigma} \in \mathtt{COIN}^{\mathsf{hy}}$,
   (a) then compute $\hat{a} := G(\hat{\sigma})$.
   (b) otherwise, output ε and halt.
4. If $\mathcal{D}^{\mathsf{sy}}_{\hat{a}}$ rejects c, then output ε and halt; otherwise, set $\hat{m} := \mathcal{D}^{\mathsf{sy}}_{\hat{a}}(c)$.
5. Set $\hat{h} := H(\hat{\sigma}, \hat{m})$.
6. If $e = \mathcal{E}^{\mathsf{asy}}_{pk}(\hat{\sigma}; \hat{h})$,
   (a) then set $\mathcal{D}^{\mathsf{hy}}_{sk}(e \parallel c) := \hat{m}$.
   (b) otherwise, set $\mathcal{D}^{\mathsf{hy}}_{sk}(e \parallel c) := \varepsilon$.
7. Return $\mathcal{D}^{\mathsf{hy}}_{sk}(e \parallel c)$.

We stress that the error symbol in Step 3 must be the same as that in Step 6. However, it is not necessary to use the same error symbol in other steps.

Theorem B.1. *Π^{hy} is $(t^{\mathsf{hy}}, q_{\mathtt{hash}}, q_{\mathtt{dec}}, \epsilon^{\mathsf{hy}})$-IND-CCA2 secure in the random oracle model where*

$$t^{\mathsf{hy}} = \min\big(t^{\mathsf{asy}}, t^{\mathsf{sy}}\big) - (q_{\mathtt{dec}} + 1)T^{\mathsf{asy}}(k) - q_{\mathtt{hash}}O(k) \quad \textit{and}$$

$$\epsilon^{\mathsf{hy}} = 2q_{\mathtt{hash}}\epsilon^{\mathsf{asy}} + \epsilon^{\mathsf{sy}} + 2q_{\mathtt{dec}}2^{-\gamma},$$

where Π^{asy} is a γ-spread $(t^{\mathsf{asy}}, \epsilon^{\mathsf{asy}})$-OWE secure asymmetric encryption scheme and Π^{sy} is a $(t^{\mathsf{sy}}, \epsilon^{\mathsf{sy}})$-OT secure bijective symmetric encryption scheme.

Proof. The proof is almost the same as the proof of Theorem 6.1, except for Game G_1. In the other games, it is enough to appropriately replace c with m. The analysis of reduction is not affected by this replacement except for Game G_1.

In Game G_1, we reply for a decryption query as follows: Let $\mathcal{G}$ and $\mathcal{H}$ be the query/answer record lists for random oracles, G and H, respectively.

For fresh query (e, c) to the decryption oracle, we proceed as follows.

1. Search tuple $((\sigma, m), h)$ in $\mathcal{H}$ such that $\sigma \in \mathtt{MSP}^{\mathsf{asy}}$ and $e = \mathcal{E}^{\mathsf{asy}}_{pk}(\sigma; h)$. If such a tuple is not recorded in $\mathcal{H}$ return ε, otherwise
2. Query oracle G on σ to obtain $a = G(\sigma)$.
3. If there is a m in the above tuples such that $\mathcal{D}^{\mathsf{sy}}_{a}(c) = m$ return m, otherwise ε.

Consider the case that "A^{hy} submits almost-valid ciphertext (e, c) to the decryption oracle, whereas there is no (σ, m, h) in $\mathcal{H}$ such that $\sigma \in \mathtt{MSP}^{\mathsf{asy}}$ and $e = \mathcal{E}^{\mathsf{asy}}_{pk}(\sigma; h)$." Let Bad_0 and Bad_1 be the events that the case occurs in G_0 and G_1, respectively. We say that (e, c) is an "almost-valid" ciphertext of Π^{hy} with respect to pk, if and only if we have

$$e = \mathcal{E}^{\mathsf{asy}}_{pk}\big(\sigma; H(\sigma, m)\big), \quad \text{where } \sigma := \mathcal{D}^{\mathsf{asy}}_{sk}(e) \text{ and } m := \mathcal{D}^{\mathsf{sy}}_{G(\sigma)}(c). \tag{B.1}$$

An almost-valid ciphertext (e, c) is valid if $\sigma \in \mathtt{MSP}^{\mathsf{asy}}$. Conditioned on Event Bad_1, the ciphertext is always rejected in Game G_0, whereas conditioned on Event Bad_0, it is not rejected in G_1 if the almost-valid ciphertext is valid.

By construction, games, G_0 and G_1 proceeds identically until the above case occurs. Hence, we have $\mathsf{Bad}_0 = \mathsf{Bad}_1$ and $S_0 \wedge \neg\mathsf{Bad}_0 = S_1 \wedge \neg\mathsf{Bad}_1$. By Lemma 7.1, we see that $\Pr[S_0] - \Pr[S_1] \leq \Pr[\mathsf{Bad}_0]$.

We now evaluate Event Bad_0. Since challenge ciphertext (e^*, c^*) is valid, we always have $\sigma^* = \mathcal{D}^{\mathsf{asy}}_{sk}(e^*)$ and $m^* = \mathcal{D}^{\mathsf{sy}}_{G(\sigma^*)}(c^*)$.

- Suppose that e is a valid ciphertext of Π^{asy} with respect to pk. Namely, there exists $\sigma \in \mathtt{MSP}^{\mathsf{asy}}$ and $r \in \mathtt{COIN}^{\mathsf{asy}}$ such that $e = \mathcal{E}^{\mathsf{asy}}_{pk}(\sigma; r)$. Then we always have $\sigma \neq \sigma^*$ or $m \neq m^*$: If $e = e^*$ (which implies $\sigma = \sigma^*$), we must have $m \neq m^*$, otherwise $c = c^*$ since Π^{sy} is *bijective*. We note that *it is not sufficient only that Π^{sy} is deterministic, because there is the case that $c \neq c^*$ with* $\mathcal{D}^{\mathsf{sy}}_{G(\sigma)}(c) = \mathcal{D}^{\mathsf{sy}}_{G(\sigma^*)}(c^*)$. If $e \neq e^*$, we must have $\sigma \neq \sigma^*$ or $m \neq m^*$, otherwise $H(\sigma, \mathcal{D}^{\mathsf{sy}}_{G(\sigma)}(m)) = H(\sigma^*, \mathcal{D}^{\mathsf{sy}}_{G(\sigma^*)}(m^*))$ and hence $e = e^* = \mathcal{E}^{\mathsf{asy}}_{pk}(\sigma; H(\sigma, m))$, which contradicts the condition.

  Hence, given ciphertext (e, c) with valid e, the value $H(\sigma, m)$ is independent of the value $H(\sigma^*, m^*)$, because $(\sigma, m) \neq (\sigma^*, m^*)$. Then consider the probability of Bad_0 on this condition. $G(\sigma)$ is uniquely determined by e (through G) and hence, m is uniquely determined by (e, c). Therefore, the conditional probability is at most $q_{\mathtt{dec}} 2^{-\gamma}$.
- On the contrary, in case e is an "invalid" ciphertext of Π^{asy} with respect to pk, we consider the possibility that the event that $\sigma = \sigma^*$ and $m = m^*$ occurs, despite the fact $(e, c) \neq (e^*, c^*)$. However, this event never occurs, because if it occurs

then $\sigma \in \mathtt{MSP}^{\mathsf{asy}}$ and $H(\sigma, m) \in \mathtt{COIN}^{\mathsf{asy}}$ (because $H(\sigma, m) = H(\sigma^*, m^*)$), which contradicts that e is invalid. So if e is invalid, (B.1) never holds and Bad_0 never occurs.

Hence, $\Pr[\mathsf{Bad}_0] \leq q_{\mathtt{dec}} 2^{-\gamma}$. Therefore, $\Pr[S_0] - \Pr[S_1] \leq q_{\mathtt{dec}} 2^{-\gamma}$. □

References

[1] M. Abdalla, M. Bellare, P. Rogaway, DHIES: An encryption scheme based on the Diffie–Hellman problem, in *IEEE P1363a*, September 2001 (2001). ANSI X9.63EC, and SECG

[2] M. Abdalla, M. Bellare, P. Rogaway, DHAES: An encryption scheme based on the Diffie–Hellman problem. Submission to IEEE P1363, November 1998. http://grouper.ieee.org/groups/1363/StudyGroup/

[3] M. Abe, R. Gennaro, K. Kurosawa, Tag-KEM/DEM: A new framework for hybrid encryption. *J. Cryptol.* **21**(1), 97–130 (2008)

[4] M. Bellare, A. Palacio, Towards plaintext-aware public-key encryption without random oracles, in *Advances in Cryptology—Asiacrypt 2004*, ed. by P.J. Lee. Lecture Notes in Computer Science, vol. 3329 (Springer, Berlin, 2004), pp. 48–62

[5] M. Bellare, P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols, in *First ACM Conference on Computer and Communication Security* (ACM, New York, 1993), pp. 62–73

[6] M. Bellare, P. Rogaway, Optimal asymmetric encryption, in *Advances in Cryptology—EUROCRYPT'94*, ed. by A.D. Santis. Lecture Notes in Computer Science, vol. 950 (Springer, Berlin, 1995), pp. 92–111

[7] M. Bellare, A. Desai, D. Pointcheval, P. Rogaway, Relations among notions of security for public-key encryption schemes, in *Advances in Cryptology—CRYPTO'98*, ed. by H. Krawczyk. Lecture Notes in Computer Science, vol. 1462 (Springer, Berlin, 1998), pp. 26–45

[8] D. Boneh, R. Canetti, S. Halevi, J. Katz, Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.* **36**(5), 1301–1328 (2007)

[9] R. Canetti, S. Halevi, J. Katz, Chosen-ciphertext security from identity based encryption, in *Advances in Cryptology—EUROCRYPT 2004*, ed. by C. Cachin, J. Camenisch. Lecture Notes in Computer Science, vol. 3027 (Springer, Berlin, 2004), pp. 207–222

[10] D. Cash, E. Kiltz, V. Shoup, The twin Diffie–Hellman problem and applications, in *EUROCRYPT*, ed. by N.P. Smart. Lecture Notes in Computer Science, vol. 4965 (Springer, Berlin, 2008), pp. 127–145

[11] R. Cramer, V. Shoup, A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack, in *Advances in Cryptology—CRYPTO'98*, ed. by H. Krawczyk. Lecture Notes in Computer Science, vol. 1462 (Springer, Berlin, 1998), pp. 13–25

[12] R. Cramer, V. Shoup, Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption, in *Advances in Cryptology—EUROCRYPT'02*. Lecture Notes in Computer Science (Springer, Berlin, 2002), pp. 45–64

[13] R. Cramer, V. Shoup, Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Comput.* **33**(1), 167–226 (2004). Early version in CRYPTO'98

[14] R. Cramer, D. Hofheinz, E. Kiltz, A twist on the Naor–Yung paradigm and its application to efficient cca-secure encryption from hard search problems, in *Theory of Cryptography—TCC 2010*. Lecture Notes in Computer Science, vol. 5978 (Springer, Berlin, 2010), pp. 146–164

[15] I. Damgård, Towards practical public key systems secure against chosen ciphertext attacks, in *Advances in Cryptology—CRYPTO'91*, ed. by J. Feigenbaum. Lecture Notes in Computer Science, vol. 576 (Springer, Berlin, 1992), pp. 445–456

[16] D. Dolev, C. Dwork, M. Naor, Non-malleable cryptography. *SIAM J. Comput.* **30**(2), 391–437 (2000) (Presented in STOC'91)

[17] Y. Frankel, M. Yung, Cryptoanalysis of the immunized LL public key systems, in *Advances in Cryptology—CRYPTO'95*, ed. by D. Coppersmith. Lecture Notes in Computer Science, vol. 963 (Springer, Berlin, 1995), pp. 287–296

[18] E. Fujisaki, T. Okamoto, Secure integration of asymmetric and symmetric encryption schemes, in *Advances in Cryptology—CRYPTO'99*, ed. by M. Wiener. Lecture Notes in Computer Science, vol. 1666 (Springer, Berlin, 1999), pp. 537–554

[19] E. Fujisaki, T. Okamoto, How to enhance the security of public-key encryption at minimum cost. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **E83-A**(1), 24–32 (2000). Early Version in PKC'99
[20] E. Fujisaki, T. Okamoto, D. Pointcheval, J. Stern, RSA-OAEP is secure under the RSA assumption, in *Advances in Cryptology—CRYPTO2001*, ed. by J. Kilian. Lecture Notes in Computer Science, vol. 2139 (Springer, Berlin, 2001), pp. 260–274
[21] D. Galindo, S. Martin, P. Morillo, J. Villar, Fujisaki-Okamoto IND-CCA hybrid encryption revisited. Technical report, IACR, May 2003. http://eprint.iacr.org/2003/107
[22] O. Goldreich, A uniform-complexity treatment of encryption and zero-knowledge. *J. Cryptol.* **6**(1), 21–53 (1993)
[23] O. Goldreich, L. Levin, A hard-core predicate for all one-way functions, in *Proceedings of the* 21*st Annual ACM Symposium on Theory of Computing (STOC'89)* (1989), pp. 25–32
[24] S. Goldwasser, S. Micali, Probabilistic encryption. *J. Comput. Syst. Sci.* **28**, 270–299 (1984)
[25] D. Hofheinz, E. Kiltz, Secure hybrid encryption from weakened key encapsulation, in *CRYPTO*, ed. by A. Menezes. Lecture Notes in Computer Science, vol. 4622 (Springer, Berlin, 2007), pp. 553–571
[26] D. Hofheinz, E. Kiltz, Practical chosen ciphertext secure encryption from factoring, in *EUROCRYPT*, ed. by A. Joux. Lecture Notes in Computer Science, vol. 5479 (Springer, Berlin, 2009), pp. 313–332
[27] M. Joye, J. Quisquater, M. Yung, On the power of misbehaving adversaries and security analysis of the original epoc, in *CT—RSA'2001*. Lecture Notes in Computer Science, vol. 2020 (Springer, Berlin, 2001), pp. 208–222
[28] J. Kilian, P. Rogaway, How to protect DES against exhaustive key search (an analysis of DESX). *J. Cryptol.* **14**(1), 17–35 (2001). Early version in CRYPTO'96
[29] E. Kiltz, K. Pietrzak, M. Stam, M. Yung, A new randomness extraction paradigm for hybrid encryption, in *Advances in Cryptology—EUROCRYPT 2009*, ed. by A. Joux. Lecture Notes in Computer Science, vol. 5479 (Springer, Berlin, 2009), pp. 590–609
[30] K. Kurosawa, Y. Desmedt, A new paradigm of hybrid encryption scheme, in *Advances in Cryptology—CRYPTO 2004*, ed. by M. Franklin. Lecture Notes in Computer Science, vol. 3152 (Springer, Berlin, 2004), pp. 426–442
[31] C. Lim, P. Lee, Another method for attaining security against adaptively chosen ciphertext attacks, in *Advances in Cryptology—CRYPTO'93*, ed. by D. Stinson. Lecture Notes in Computer Science, vol. 773 (Springer, Berlin, 1993)
[32] Y. Lindell, A simpler construction of cca2-secure public-key encryption under general assumptions, in *Advances in Cryptology—EUROCRYPT'03*, ed. by E. Biham. Lecture Notes in Computer Science, vol. 2656 (Springer, Berlin, 2003), pp. 241–254
[33] M. Naor, M. Yung, Public-key cryptosystems provably secure against chosen ciphertext attacks, in *Proceedings of the* 21*st Annual ACM Symposium on Theory of Computing (STOC'90)* (1990), pp. 427–437
[34] T. Okamoto, D. Pointcheval, REACT: Rapid enhanced-security asymmetric cryptosystem transform, in *CT—RSA'2001*. Lecture Notes in Computer Science, vol. 2020 (Springer, Berlin, 2001), pp. 159–175
[35] T. Okamoto, S. Uchiyama, A new public-key cryptosystem as secure as factoring, in *Advances in Cryptology—EUROCRYPT'98*, ed. by K. Nyberg. Lecture Notes in Computer Science, vol. 1403 (Springer, Berlin, 1998), pp. 308–318
[36] T. Okamoto, S. Uchiyama, E. Fujisaki, EPOC: Efficient probabilistic public-key encryption. Submission to IEEE P1363. http://info.isl.ntt.co.jp/epoc
[37] C. Peikert, B. Waters, Lossy trapdoor functions and their applications, in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC'08)* (2008)
[38] D.H. Phan, D. Pointcheval, OAEP 3-round: A generic and secure asymmetric encryption padding, in *Advances in Cryptology—Asiacrypt 2004*, ed. by P.J. Lee. Lecture Notes in Computer Science, vol. 3329 (Springer, Berlin, 2004), pp. 63–78
[39] D. Pointcheval, Chosen-ciphertext security for any one-way cryptosystem, in *3rd International Workshop on Practice and Theory in Public Key Cryptography—PKC'00*, ed. by H. Imai, Y. Zheng. Lecture Notes in Computer Science, vol. 1751 (Springer, Berlin, 2000), pp. 129–146
[40] C. Rackoff, D. Simon, Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack, in *Advances in Cryptology—CRYPTO'91*, ed. by J. Feigenbaum. Lecture Notes in Computer Science, vol. 576 (Springer, Berlin, 1992), pp. 433–444
[41] A. Rosen, G. Segev, Chosen-ciphertext security via correlated products, in *Theory of Cryptography—TCC 2009*, ed. by O. Reingold. Lecture Notes in Computer Science, vol. 5444 (Springer, Berlin, 2009), pp. 419–436

[42] A. Sahai, Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security, in *Proceedings of the 40th IEEE Annual Symposium on Foundations of Computer Science (FOCS'99)* (1999), pp. 543–553

[43] K. Sakurai, T. Takagi, A reject timing attack on an ind-cca2 public-key cryptosystem, in *ICISC'02*. Lecture Notes in Computer Science, vol. 2587 (Springer, Berlin, 2001), pp. 359–373

[44] V. Shoup, OAEP Reconsidered, in *Advances in Cryptology—CRYPTO2001*, ed. by J. Kilian. Lecture Notes in Computer Science, vol. 2139 (Springer, Berlin, 2001), pp. 239–259

[45] V. Shoup, A proposal for an ISO standard for public key encryption. Technical report, Cryptology ePrint Archive, Report 2001/112, December 2001

[46] H. Wee, Efficient chosen-ciphertext security via extractable hash proofs, in *CRYPTO*, ed. by T. Rabin. Lecture Notes in Computer Science, vol. 6223 (Springer, Berlin, 2010), pp. 314–332

[47] Y. Zheng, J. Seberry, Immunizing public key cryptosystems against chosen ciphertext attacks. *J. Sel. Areas Commun.* 11(5) (1993)