PAPER *Special Section on Cryptography and Information Security*

Randomness Leakage in the KEM/DEM Framework*

Hitoshi NAMIKI[†], ***Nonmember***, **Keisuke TANAKA**[††], ***and*** **Kenji YASUNAGA**[†††a)], ***Members***

SUMMARY Recently, there have been many studies on constructing cryptographic primitives that are secure even if some secret information leaks. In this paper, we consider the problem of constructing public-key encryption schemes that are resilient to leaking the randomness used in the encryption algorithm. In particular, we consider the case in which public-key encryption schemes are constructed from the KEM/DEM framework, and the leakage of randomness in the encryption algorithms of KEM and DEM occurs independently. For this purpose, we define a new security notion for KEM. Then we provide a generic construction of a public-key encryption scheme that is resilient to randomness leakage from any KEM scheme satisfying this security. Also we construct a KEM scheme that satisfies the security from hash proof systems.

key words: *leakage resilient cryptography, public-key encryption, KEM/DEM framework, hash proof system*

1. Introduction

Recently, many studies have been devoted to construct encryption schemes that are secure even if some of the secret information leaks [1]–[5], [10], [15], [16]. In [1], [2], [5], [10], [15], they mainly considered the case of leaking a secret key. In particular, studies in [1], [2], [15] considered the case that any of the secret-key information leaks, and the restriction is only the amount of the leaked information. This leakage model captures many realistic attacks including side-channel attack and cold-boot attack [11]. Naor and Segev [15] also studied the case of leaking the randomness used in the key generation algorithm. They proved that their proposed encryption scheme, which is resilient to the secret-key leakage, is also resilient to the leakage of the randomness used in the key generation algorithm.

Bellare et al. [4] considered the case of leaking the randomness used in the encryption algorithm. Their definition of leaking the randomness is different from those of other studies of information leakage. They considered the case that random strings are sampled from not a uniformly random distribution but an entropically guaranteed distribution.

In this work, we investigate the possibility of constructing public-key encryption schemes that are secure even if the randomness information used in the encryption algorithm leaks. The restriction we consider is only the amount of the leaked information.

First we define the security notions of public-key encryption in the presence of the leakage of the randomness used in the encryption algorithm. The definitions are similar to those of the secret-key leakage introduced in [1], [15]. We define two randomness-leakage attacks, *a priori randomness-leakage attack* and *a posteriori randomness-leakage attack*. In the a priori randomness-leakage attack, the adversary can obtain the leakage information on the randomness before she receives a public key. In the a posteriori randomness-leakage attack, the adversary can obtain the leakage information after receiving the public key. Then we show that a secure public-key encryption scheme against a priori randomness-leakage attack can be constructed from any secure public-key encryption scheme. This is proved by a similar argument to the case of key leakage in [15]. However, for the a posteriori randomness-leakage attack, we show that no public-key encryption scheme can achieve the security. This situation is contrast to the case of secret-key leakage. Indeed, it is shown that a secure scheme for key leakage can be constructed from any hash proof system [15]. The results are summarized in Table 1, in which we compare the results of randomness leakage with that of key leakage.

Next, we focus on public-key encryption schemes based on the framework of key-encapsulation mechanism (KEM) and data-encapsulation mechanism (DEM). Since no scheme can achieve the randomness-leakage resilience if the leakage occurs after the adversary receives the public key, we restrict the way of leakage as follows. (1) The leakage of randomness used in KEM and that in DEM occurs independently, and (2) the leakage of randomness used in DEM occurs after the adversary chose two challenge messages. The situation of the first condition arises when the computation of KEM and that of DEM are implemented by two independent computer chips. The second condition naturally arises since the computation of DEM, which contains the information on a message, cannot be performed before choosing the message to be encrypted. Regarding the leakage amount, we restrict the amount of the randomness leakage only for the encryption of KEM, and not for DEM. Namely, the adversary can learn the entire random bits used

Manuscript received March 28, 2013.
Manuscript revised July 10, 2013.

[†] The author was with the Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, 152-8552 Japan.

[††] The author is with the Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, 152-8552 Japan.

[†††] The author is with the Faculty of Electrical and Computer Engineering, the Institute of Science and Engineering, Kanazawa University, Kanazawa-shi, 920-1192 Japan.

[*] A preriminary version of this paper appeared in the Proceedings of the Fifth International Conference on Provable Security (ProvSec 2011).

a) E-mail: yasunaga@se.kanazawa-u.ac.jp

DOI: 10.1587/transfun.E97.A.191

Table 1 Comparison between key leakage and randomness leakage.

Timing of leakage	Key leakage	Randomness leakage
Before receiving the public key	IND-CPA PKE [15]	IND-CPA PKE (Sec. 3.1)
After receiving the public key	Hash Proof Systems [15]	Impossible (Sec. 3.2)
After receiving the ciphertext	Impossible [1]	Impossible (Sec. 3.2)

Table 2 Information leakage in public-key encryption schemes.

References	Leakage information	Assumption	Timing of leakage
[1]	Secret key	LWE	After receiving the public key
[15]	Secret key	HPS	After receiving the public key
[4]	Randomness	Lossy TDF	Before receiving the public key
This work	Randomness*	HPS	After receiving the public key

* The leakage model is the KEM/DEM model defined in Definition 7. As presented in Table 1, no PKE scheme is secure against leakage attacks after receiving the public key when considering the model defined in Definition 6.

in the encryption of DEM. Note that we allow an encryption algorithm of DEM to be randomized, while it is usually deterministic.

To construct a public-key encryption scheme secure against randomness leakage attack, we define a new security notion, called *entropic security* for KEM. A KEM scheme is entropically secure if there are fake public-keys such that the distribution of real public-keys and that of fake public-keys are computationally indistinguishable, and if a fake key is used instead of a real key, the symmetric key is statistically close to some high entropy distribution. Then, we provide a generic construction of a public-key encryption scheme that is resilient to randomness leakage from any entropically secure KEM scheme.

Also we construct an entropically secure KEM scheme from hash proof systems [6], [15]. The scheme can be seen as a variant of KEM scheme of Naor and Segev [15], which is secure against "secret-key" leakage attack.

In Table 2, we summarize the results of information leakage on public-key encryption schemes. Note that the scheme proposed in this work is the first scheme that achieves the security against randomness leakage after receiving the public key.

1.1 Related Work

The key-leakage security in which the adversary can learn any information on the secret key was first formalized by Akavia, Goldwasser, and Vaikuntanathan [1]. In addition, they showed that Regev's lattice-based scheme is resilient to the key leakage. Naor and Segev [15] extended the notion of [1], and they proposed a general construction of public-key encryption schemes that are resilient to key leakage based on universal hash proof systems. In addition, they applied the notion of leakage to the randomness used in the key-generation algorithm. Alwen, Dodis, and Wichs [2] constructed varieties of key-leakage resilient public-key cryptosystems, such as identification, signature, and authenticated key agreement. See [3], [16] for surveys of leakage-resilient cryptography.

Recently, Halevi and Lin [12] have studied a realizable security of public-key encryption schemes in which the leakage occurs after the adversary receives the ciphertext. In particular, they bypass the impossibility of the ciphertext-dependent leakage by assuming that the secret key consists of two parts, and the leakage of the two parts occurs independently. Their approach of "split-state" model is similar to our approach, and used in several other studies in leakage-resilient cryptography [7], [10], [14].

There are several studies related to the leakage of the randomness used in the encryption algorithm. Bellare et al. [4] considered the situation in which the randomness used in the encryption algorithm is not uniformly random, but is entropically guaranteed. They introduced a security notion such that even if the randomness is not chosen uniformly at random, the scheme is secure as long as the joint distribution of the message and the randomness has high entropy. Note that the distribution of the randomness is chosen without using the information on the public key, which is different from the setting of our work.

Kamara and Katz [13] studied another type of randomness leakage in symmetric-key encryption. In their setting, the adversary can control the randomness of the ciphertext except that of the challenge ciphertext.

There are other formalization of information leakage. Dizembowski and Pietrzak [10] considered the key leakage under the assumption that only the computation leaks information, and constructed leakage-resilient stream-ciphers. Dodis, Tauman Kalai, and Lovett [9] studied symmetric-key encryption schemes under key-leakage attack. They considered the leakage of the form $f(sk)$, where sk is the secret key and f is any exponentially-hard one-way function, and do not restrict the entropy of the secret key.

2. Preliminaries

In this section, we present notions, definitions, and tools that are used in our constructions. Let n be the security parameter on all of the schemes in this paper, and U_t the uniform distribution over $\{0, 1\}^t$, where $t \in \mathbb{N}$. For a distribution X, we write $x \leftarrow X$ to indicate that x is chosen according to X. For a finite set Y, we write $y \leftarrow Y$ to indicate that y is chosen from Y uniformly at random. We say an algorithm is PPT if it runs by a probabilistic polynomial-time Turing machine.

2.1 Randomness Extraction

The *statistical distance* between two random variables X and Y over a finite domain Ω is $\Delta(X, Y) := \frac{1}{2}\sum_{\omega\in\Omega}|\Pr[X = \omega] - \Pr[Y = \omega]|$. We say that two variables are *ϵ-close* if their statistical distance is at most ϵ. The *min-entropy* of a random variable X is $H_\infty(X) := -\log(\max_x \Pr[X = x])$. The min-entropy is a standard notion of entropy used in cryptography since it measures the worst case predictability of X. We also use the *average min-entropy* defined as follows:

$$\tilde{H}_\infty(X|Y) := -\log\left(E_{y\leftarrow Y}\left[2^{-H_\infty(X|Y=y)}\right]\right).$$

The average min-entropy represents the optimal predictability of X, given knowledge of Y. The following lemma was proved by Dodis, Ostrovsky, Reyzin, and Smith [8], which will be used in this paper.

Lemma 1: Let $r \in \mathbb{R}$. If Y has 2^r possible values and Z is any random variable, then for a random variable X, $\tilde{H}_\infty(X|(Y, Z)) \geq \tilde{H}_\infty(X|Z) - r$.

We use a strong randomness extractor as a main tool in our constructions. The following definition naturally generalizes the standard definition of strong extractors to the setting of the average min-entropy.

Definition 1: A function $\mathsf{Ext} : \{0, 1\}^k \times \{0, 1\}^t \to \{0, 1\}^m$ is an *average-case (n, ϵ)-strong extractor* if for all the pairs of random variables (X, I) such that $X \in \{0, 1\}^k$, and $\tilde{H}_\infty(X|I) \geq n$, it holds that

$$\Delta\left((\mathsf{Ext}(X, U_t), U_t, I), (U_m, U_t, I)\right) \leq \epsilon.$$

Dodis et al. proved the following variant of the leftover hash lemma. Any family of pairwise independent hash functions is indeed an average-case strong extractor [8].

Lemma 2: Let X, Y be random variables such that $X \in \{0, 1\}^n$ and $\tilde{H}_\infty(X|Y) \geq k$. Let $\mathcal{H}$ be a family of pairwise independent hash functions from $\{0, 1\}^n$ to $\{0, 1\}^m$. Then for $h \in \mathcal{H}$ chosen uniformly at random, it holds that

$$\Delta((Y, h, h(X)), (Y, h, U_m)) \leq \epsilon$$

as long as $m \leq k - 2\log(1/\epsilon)$.

2.2 The KEM/DEM Framework

We present the framework of key-encapsulation mechanism (KEM) and data-encapsulation mechanism (DEM). The KEM/DEM paradigm is a simple way of constructing efficient and practical public-key encryption schemes. KEM is used as public-key encryption used for encrypting a random symmetric key K together with its ciphertext. The symmetric key is used for encrypting the message using DEM. A formal definition of KEM and DEM is given as follows.

Definition 2: Key encapsulation mechanism is a tuple of PPT algorithms $KEM = (\mathsf{KEM.Gen}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec})$ such that

KEM.Gen : On input a security parameter 1^n, output a pair of keys (pk, sk).

KEM.Enc : On input a public key pk and a random string r from some underlying randomness space, output a ciphertext c and a symmetric key K.

KEM.Dec : On input a secret key sk and a ciphertext c, output a symmetric key K.

It is required that for any $(pk, sk) \leftarrow \mathsf{KEM.Gen}(1^n)$ and any random string r,

$$\mathsf{KEM.Dec}(sk, c) = K,$$

where $(c, K) \leftarrow \mathsf{KEM.Enc}(pk, r)$.

Definition 3: Data encapsulation mechanism is a tuple of PPT algorithms $DEM = (\mathsf{DEM.Gen}, \mathsf{DEM.Enc}, \mathsf{DEM.Dec})$ such that

DEM.Gen : On input a security parameter 1^n, output a symmetric key K.

DEM.Enc : On input a symmetric key K, a message m, and a random string r, output a ciphertext c.

DEM.Dec : On input a symmetric key K and a ciphertext c, output a message m.

Note that we define DEM.Enc as a randomized algorithm, while it is usually deterministic. We provide a KEM/DEM construction in Sect. 5 in which the encryption algorithm of DEM is randomized.

In this paper, we only consider the case in which a public-key encryption scheme is constructed from KEM/DEM paradigm. KEM is used for exchanging a symmetric key K. The message is encrypted using DEM with the symmetric key K. Thus, a public-key encryption scheme can be written as a tuple of five PPT algorithms (KEM.Gen, KEM.Enc, KEM.Dec, DEM.Enc, DEM.Dec).

2.3 Hash Proof Systems

Hash proof systems were introduced by Cramer and Shoup [6]. We briefly review the presentation in [15], where hash proof systems are viewed as key encapsulation mechanisms.

In hash proof systems, ciphertexts are generated in two modes. In the first mode, *valid ciphertexts* are generated, where encapsulated key is well-defined and decapsulated with the secret key. Additionally, the process of generating a valid ciphertext produces a *witness* to the fact that the ciphertext is valid. In general, the randomness for generating a valid ciphertext can be a witness for that ciphertext, and we will use this fact in our construction. In the second mode, *invalid ciphertexts* are generated, and essentially contain no information on the encapsulated key. The computational requirement is that the two modes are computationally indistinguishable.

2.3.1 Smooth Projective Hashing

Let $\mathcal{SK}$ and $\mathcal{PK}$ be the domains of secret and public keys, $\mathcal{K}$ the encapsulated key space, C the ciphertext space, and $\mathcal{V} \subset C$ the valid ciphertext space. Let $\Lambda = \{\Lambda_{sk} : C \to \mathcal{K}\}$ be a collection of hash functions. Then, Λ is called *projective* if there is a projection $\mu : \mathcal{SK} \to \mathcal{PK}$ such that for all $v \in \mathcal{V}$, $sk_1, sk_2 \in \mathcal{SK}$ satisfying $\mu(sk_1) = \mu(sk_2)$, it holds that $\Lambda_{sk_1}(v) = \Lambda_{sk_2}(v)$. A projective hash function is called *δ-smooth* if for all $c' \in C \setminus \mathcal{V}$, it holds that

$$\Delta((pk, c', \Lambda_{sk}(c')), (pk, c', K)) \le \delta,$$

where $sk \in \mathcal{SK}$ and $K \in \mathcal{K}$ are sampled uniformly at random, and $pk = \mu(sk)$.

2.3.2 Hash Proof System

Definition 4: A hash proof system is a tuple of three polynomial-time algorithms $\mathsf{HPS} = (\mathsf{Param}, \mathsf{Pub}, \mathsf{Priv})$ such that

Param**:** On input a security parameter 1^n, output the description $(\mathcal{SK}, \mathcal{PK}, C, \mathcal{V}, \mathcal{K}, \Lambda, \mu)$.

Pub**:** On input a public key $pk = \mu(sk)$, a valid ciphertext $c \in \mathcal{V}$, and a witness w of the fact that $c \in \mathcal{V}$, output the encapsulated key $K = \Lambda_{sk}(c)$. We assume that Pub is a deterministic algorithm, and that the witness w is randomness for sampling c from $\mathcal{V}$.

Priv**:** On input a secret key $sk \in \mathcal{SK}$ and a ciphertext c, output the encapsulated key $K = \Lambda_{sk}(c)$, which is the same as the key obtained by $\mathsf{Pub}(\mu(sk), c, w)$, where w is a witness of the fact that $c \in \mathcal{V}$. We assume that Priv is a deterministic algorithm.

If a collection Λ of hash functions is δ-smooth, then we say that the hash proof system has *δ-smoothness*.

As a computational problem, we require the *subset membership problem* is hard in HPS. Formally, we require that for any PPT algorithm A,

$$\mathsf{Adv}^{\mathsf{SMP}}_{\mathsf{HPS},A}(n) := \left| \Pr_{c_0 \leftarrow \mathcal{V}}[A(F, c_0) = 1] - \Pr_{c_1 \leftarrow C \setminus \mathcal{V}}[A(F, c_1) = 1] \right|$$

is negligible in n, where $F = (\mathcal{SK}, \mathcal{PK}, C, \mathcal{V}, \mathcal{K}, \Lambda, \mu)$ is generated by $\mathsf{Param}(1^n)$.

3. Randomness Leakage in Public-Key Encryption

Akavia et al. [1] introduced the security of public-key encryption schemes in the presence of secret-key leakage. We formalize the security in the presence of randomness leakage based on their notion. In particular, we consider the leakage of the randomness used in the encryption algorithm.

We define two randomness-leakage attacks, *a priori randomness-leakage attack* and *a posteriori randomness-leakage attack*. In the a priori randomness-leakage attack, the adversary can have access to the leakage oracle before she obtains a public key. In the a posteriori randomness-leakage attack, the adversary can have access to the leakage oracle after she obtains the public key.

3.1 A Priori Randomness-Leakage Attack

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme. Namely, on input a security parameter, Gen outputs a pair of a public key and a secret key. On input a public key and a message, Enc outputs a ciphertext. On input a secret key and a ciphertext, Dec outputs a message. Let $\ell(n)$ be the length of the randomness used in Enc, where n is the security parameter. The leakage oracle, denoted by $\mathsf{RandLeak}(r)$, takes as input a function $f : \{0,1\}^{\ell(n)} \to \{0,1\}^*$ and outputs $f(r)$, where r is the randomness used in Enc. We call the adversary A is an *a priori $\lambda(n)$-randomness-leakage adversary* if the sum of the output length of $\mathsf{RandLeak}$ that A queries is at most $\lambda(n)$, and f is chosen by A before she receives the public key. Note that, although we may consider adaptive leakage, in which the adversary can access to the leakage oracle adaptively, it does not affect our definitions of randomness leakage as discussed in [1].

Definition 5: A public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is *a priori $\lambda(n)$-randomness-leakage resilient* if for any PPT a priori $\lambda(n)$-randomness-leakage adversary $A = (A_1, A_2, A_3)$, it holds that

$$\mathsf{Adv}^{\mathsf{apriori}}_{\Pi,A}(n) := \left| \Pr\left[\mathsf{Expt}^{\mathsf{apriori}}_{\Pi,A}(0) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{apriori}}_{\Pi,A}(1) = 1\right] \right|$$

is negligible in n, where the experiment $\mathsf{Expt}^{\mathsf{apriori}}_{\Pi,A}(b)$ is defined as follows.

1. $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$.
2. Choose $r \leftarrow U_{\ell(n)}$.
3. $st_1 \leftarrow A_1^{\mathsf{RandLeak}(r)}(1^n)$.
4. $(m_0, m_1, st_2) \leftarrow A_2(pk, st_1)$ such that $|m_0| = |m_1|$.
5. $c \leftarrow \mathsf{Enc}(pk, m_b, r)$.
6. $b' \leftarrow A_3(c, st_2)$.
7. Output b'.

We provide a construction of public-key encryption that is resilient to a priori randomness-leakage attack based on any IND-CPA secure scheme. The construction is similar to that of [1] for the case of the secret-key leakage.

Construction 1: Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme, $\ell(n)$ the length of the random string used in Enc, and $\mathsf{Ext} : \{0,1\}^{k(n)} \times \{0,1\}^{t(n)} \to \{0,1\}^{\ell(n)}$ an average-case extractor. The scheme $\Pi^* = (\mathsf{Gen}^*, \mathsf{Enc}^*, \mathsf{Dec}^*)$ is defined as follows.

Gen^***:** On input 1^n, choose $s \leftarrow U_{t(n)}$, compute $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$, and output $PK = (pk, s)$ and $SK = sk$.

Enc^***:** On input a message m and a public key $PK = (pk, s)$, choose $r \leftarrow U_{k(n)}$ and output $\mathsf{Enc}(pk, m, \mathsf{Ext}(r, s))$.

Dec^***:** On input a ciphertext c and a secret key $SK = sk$, output $\mathsf{Dec}(sk, c)$.

Theorem 1: Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure public-key encryption scheme and $\mathsf{Ext} : \{0,1\}^{k(n)} \times \{0,1\}^{t(n)} \rightarrow \{0,1\}^{m(n)}$ an average-case $(k(n) - \lambda(n), \epsilon(n))$-strong extractor for some negligible function $\epsilon(n)$. Then, the encryption scheme Π^* is a priori $\lambda(n)$-randomness-leakage resilient.

Proof. We show that for any adversary A, there exists an adversary A' such that

$$\mathsf{Adv}^{\mathsf{apriori}}_{\Pi^*,A}(n) \leq \mathsf{Adv}^{\mathsf{CPA}}_{\Pi,A'}(n) + 2\epsilon(n),$$

where the $\mathsf{Adv}^{\mathsf{CPA}}_{\Pi,A'}(n)$ the advantage of A' in the IND-CPA game with Π. Consider the following experiment $\mathsf{Expt}_{\Pi,A}(b)$:

1. $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$, choose $r \leftarrow U_{k(n)}$, $s \leftarrow U_{t(n)}$, and $z \leftarrow U_{m(n)}$. Let $PK = (pk, s)$ and $SK = sk$.
2. $st_1 \leftarrow A_1^{\mathsf{RandLeak}(r)}(1^n)$.
3. $(m_0, m_1, st_2) \leftarrow A_2(PK, st_1)$ such that $|m_0| = |m_1|$.
4. $c \leftarrow \mathsf{Enc}(pk, m_b, z)$.
5. $b' \leftarrow A_3(c, st_2)$.
6. Output b'.

From Definition 5 and the triangle inequality, it follows that

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{apriori}}_{\Pi^*,A}(n)\\
&= \left|\Pr\left[\mathsf{Expt}^{\mathsf{apriori}}_{\Pi^*,A}(0) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{apriori}}_{\Pi^*,A}(1) = 1\right]\right|\\
&\leq \left|\Pr\left[\mathsf{Expt}^{\mathsf{apriori}}_{\Pi^*,A}(0) = 1\right] - \Pr\left[\mathsf{Expt}_{\Pi,A}(0) = 1\right]\right|\\
&\quad + \left|\Pr\left[\mathsf{Expt}_{\Pi,A}(0) = 1\right] - \Pr\left[\mathsf{Expt}_{\Pi,A}(1) = 1\right]\right|\\
&\quad + \left|\Pr\left[\mathsf{Expt}_{\Pi,A}(1) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{apriori}}_{\Pi^*,A}(1) = 1\right]\right|.
\end{aligned}$$

The experiment $\mathsf{Expt}_{\Pi,A}(b)$ is identical to the experiment $\mathsf{Expt}^{\mathsf{apriori}}_{\Pi^*,A}(b)$, except for the fact that Enc uses a truly random input z, not $\mathsf{Ext}(r, s)$. Note that, from Lemma 1, given the information of $f(r)$, the average min-entropy of r is at least $(k - \lambda)$. Therefore the average-case strong extractor guarantees that the statistical distance between the view of the adversary in these two experiments is at most $\epsilon(n)$. This implies that $\left|\Pr\left[\mathsf{Expt}_{\Pi,A}(b) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{apriori}}_{\Pi^*,A,F}(b) = 1\right]\right| \leq \epsilon(n)$ for $b \in \{0,1\}$. Since $\mathsf{Expt}_{\Pi,A}(b)$ is the same as the IND-CPA experiment, we can construct the IND-CPA adversary A' for which $\left|\Pr\left[\mathsf{Expt}_{\Pi,A}(0) = 1\right] - \Pr\left[\mathsf{Expt}_{\Pi,A}(1) = 1\right]\right| \leq \mathsf{Adv}^{\mathsf{CPA}}_{\Pi,A'}(n)$. □

3.2 A Posteriori Randomness-Leakage Attack

In this section, we define a posteriori randomness-leakage attack. Consequently, we show that there is no public-key encryption scheme that is a posteriori randomness-leakage resilient even if the leakage information is only one bit.

We define the security in a similar way as in Definition 5. An adversary A is called *a posteriori $\lambda(n)$-randomness-leakage adversary* if the sum of the output lengths of $\mathsf{RandLeak}$ is at most $\lambda(n)$, and a leakage function f is chosen by A before receiving the challenge ciphertext.

Definition 6: A public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is *a posteriori $\lambda(n)$-randomness-leakage resilient* if for any PPT a posteriori $\lambda(n)$-randomness-leakage adversary $A = (A_1, A_2)$, it holds that

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{aposteriori}}_{\Pi,A}(n)\\
&\coloneqq \left|\Pr\left[\mathsf{Expt}^{\mathsf{aposteriori}}_{\Pi,A}(0)=1\right]-\Pr\left[\mathsf{Expt}^{\mathsf{aposteriori}}_{\Pi,A}(1)=1\right]\right|
\end{aligned}$$

is negligible in n. The experiment $\mathsf{Expt}^{\mathsf{aposteriori}}_{\Pi,A}(b)$ is defined as follows:

1. $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$.
2. Choose $r \leftarrow U_{\ell(n)}$.
3. $(m_0, m_1, st_1) \leftarrow A_1^{\mathsf{RandLeak}(r)}(pk)$ such that $|m_0| = |m_1|$.
4. $c \leftarrow \mathsf{Enc}(pk, m_b, r)$.
5. $b' \leftarrow A_2(c, st_1)$.
6. Output b'.

We show that no public-key encryption scheme achieves Definition 6. We construct an adversary A' that breaks the a posteriori randomness-leakage resilience, where $\lambda(n) = 1$. The strategy of A' is as follows. First, A' makes two challenge messages m_0, m_1 arbitrary, and randomly chooses $1 \leq i \leq d(n)$, where $d(n)$ is the maximum length of the possible ciphertexts. Then A' asks the leakage oracle with $f(\cdot) = \{$the i-th bit of the output of $\mathsf{Enc}(pk, m_1, \cdot)\}$. After A' receives the challenge ciphertext c, she checks whether the i-th bit of c and $f(r)$ are the same or not. If they are, she outputs 1, and otherwise outputs 0. There exists at least one position where the ciphertext of m_0 and that of m_1 are different because of the correctness of the scheme. If i is such a position, then A' can correctly predict the challenge message. The probability it occurs is at least $1/d(n)$, which is a lower bound of the probability $\Pr\left[\mathsf{Expt}^{\mathsf{aposteriori}}_{\Pi,A'}(0) = 0\right]$. On the other hand, $\Pr\left[\mathsf{Expt}^{\mathsf{aposteriori}}_{\Pi,A'}(1) = 1\right] = 1$. Thus,

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{aposteriori}}_{\Pi,A}(n)\\
&=\left|\Pr\left[\mathsf{Expt}^{\mathsf{aposteriori}}_{\Pi,A}(0)=1\right]-\Pr\left[\mathsf{Expt}^{\mathsf{aposteriori}}_{\Pi,A}(1)=1\right]\right|\\
&=\left|\left(1-\Pr\left[\mathsf{Expt}^{\mathsf{aposteriori}}_{\Pi,A'}(0)=0\right]\right)-1\right|\\
&\geq \frac{1}{d(n)},
\end{aligned}$$

which is non-negligible.

4. Randomness Leakage in KEM/DEM

In this section, we define randomness-leakage attack for

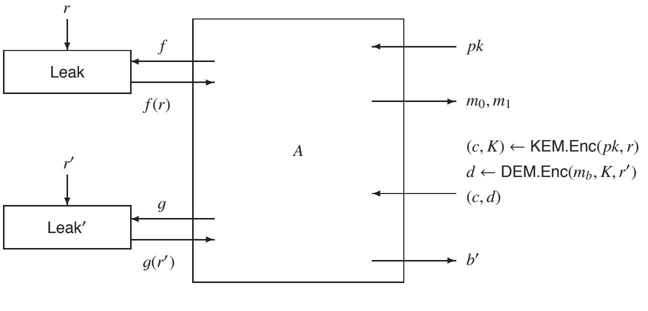

r
Leak
f
$f(r)$
r'
Leak′
g
$g(r')$
A
pk
m_0, m_1
$(c, K) \leftarrow \mathsf{KEM.Enc}(pk, r)$
$d \leftarrow \mathsf{DEM.Enc}(m_b, K, r')$
(c, d)
b'

Fig. 1 Experiment $\mathsf{Expt}^{\mathsf{RandLeak}}_{\Pi,A}(b)$.

KEM/DEM-based public-key encryption schemes. As discussed in Sect. 3, there is no public-key encryption that achieves a posteriori randomness-leakage. Therefore, we restrict the leakage of randomness such that the leaking of random bits used in KEM.Enc and DEM.Enc occur independently. Also when the adversary chooses two challenge messages, she is not allowed to access to the leakage information of random bits in DEM.Enc.

We describe the formal definition of the randomness-leakage attack in KEM/DEM. The randomness-leakage oracle for KEM.Enc, denoted by Leak, takes as input a function $f : R \to \{0,1\}^*$ and outputs $f(r)$, where R is the domain of the randomness used in KEM.Enc and r is the random bits generated in KEM.Enc. We restrict the function f to be efficiently computable. The leakage oracle for DEM.Enc, denoted by Leak′, takes as input a function $g : R' \to \{0,1\}^*$ and output $g(r')$, where R' is the domain of the randomness used in DEM.Enc and r' is the random bits generated in DEM.Enc. We restrict the amount of the leaked bits for r but not for r'. Namely, the adversary can learn the entire random bits r', which are generated in DEM.Enc. We call an adversary A is a *$\lambda(n)$-randomness-leakage adversary* if the sum of the output length of Leak that A queries is at most $\lambda(n)$.

Definition 7: A public-key encryption scheme Π = (KEM.Gen, KEM.Enc, KEM.Dec, DEM.Enc, DEM.Dec) is *IND-CPA secure against $\lambda(n)$-randomness-leakage attack* if for any PPT $\lambda(n)$-randomness-leakage adversary $A = (A_1, A_2)$ it holds that,

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{RandLeak}}_{\Pi,A}(n)\\ &:= \left|\Pr[\mathsf{Expt}^{\mathsf{RandLeak}}_{\Pi,A}(0)=1] - \Pr[\mathsf{Expt}^{\mathsf{RandLeak}}_{\Pi,A}(1)=1]\right|\end{aligned}$$

is negligible in n, where $\mathsf{Expt}^{\mathsf{RandLeak}}_{\Pi,A}(b)$ is defined as follows.

1. $(pk, sk) \leftarrow \mathsf{KEM.Gen}(1^n)$.
2. Choose $r \in R$ uniformly at random.
3. $(c, K) \leftarrow \mathsf{KEM.Enc}(pk, r)$.
4. $(m_0, m_1, st_1) \leftarrow A_1^{\mathsf{Leak}(r)}(pk)$ such that $|m_0| = |m_1|$.
5. Choose $r' \in R'$ uniformly at random.
6. $d \leftarrow \mathsf{DEM.Enc}(m_b, K, r')$.
7. $b' \leftarrow A_2^{\mathsf{Leak}'(r')}(c, d, st_1)$.
8. Output b'.

Note that while the randomness of KEM is leaked when the adversary chooses the challenge messages, that of DEM is leaked only after submitting the challenge messages. This captures a natural situation where the ciphertext of KEM may (and can) be generated before a message to be encrypted is determined, and the ciphertext of DEM is generated after a message to be encrypted is determined.

5. Randomness-Leakage Resilient Schemes from Entropically-Secure KEM

In this section, we first define a new security notion for KEM. Then, we construct a public-key encryption scheme that is IND-CPA secure against randomness-leakage attack from any KEM scheme satisfying this security.

For a KEM scheme, we require that the symmetric key K still has high average min-entropy even if the adversary knows the public key pk, the ciphertext c, and any partial information $f(r)$ of the random string r. We consider a distribution PK^* that is computationally indistinguishable from the real distribution of pk, where pk is generated from the key generation algorithm of KEM. Given $pk^* \in PK^*$, c^*, and $f(pk^*, r)$, we require that K^* is statistically close to some distribution that has enough entropy, where (c^*, K^*) is generated according to $\mathsf{KEM.Enc}(pk^*, r)$ and f is an arbitrary efficiently computable function whose output length is restricted.

Definition 8: A KEM scheme (KEM.Gen, KEM.Enc, KEM.Dec) is *$(\kappa(n), \epsilon(n))$-entropically secure against $\lambda(n)$-randomness-leakage attack* if

1. there exists an efficiently samplable distribution PK^* that is computationally indistinguishable from the distribution $\{pk \,|\, (pk, sk) \leftarrow \mathsf{KEM.Gen}(1^n)\}$, and
2. there is a distribution K' such that

$$\tilde{H}_\infty(K' | pk^*, c^*, f(pk^*, r)) \geq \kappa(n)$$

and

$$\Delta((pk^*, c^*, K^*, f(pk^*, r)), (pk^*, c^*, K', f(pk^*, r))) \leq \epsilon(n),$$

where $pk^* \leftarrow PK^*$, $(c^*, K^*) \leftarrow \mathsf{KEM.Enc}(pk^*, r)$, $r \in R$ is sampled uniformly at random, and f is an arbitrary efficiently computable function whose output length is at most $\lambda(n)$.

We can construct a public-key encryption scheme secure against randomness-leakage attack from any KEM scheme satisfying Definition 8.

Theorem 2: Let $\mathsf{Ext} : G \times \{0,1\}^t \to \{0,1\}^m$ be an average-case $(\kappa(n) - \lambda(n), \epsilon_1(n))$-strong extractor, and (KEM.Gen, KEM.Enc, KEM.Dec) a KEM scheme that is $(\kappa(n)-\lambda(n), \epsilon_2(n))$-entropically secure against $\lambda(n)$-randomness-leakage attack for some negligible functions $\epsilon_1(\cdot)$ and $\epsilon_2(\cdot)$. Then, the following scheme Π^* = (KEM.Gen*, KEM.Enc*, KEM.Dec*, DEM.Enc*, DEM.Dec*) is a public-key encryption scheme that is IND-CPA secure against $\lambda(n)$-randomness-leakage attack.

Construction 2: The algorithms $\mathsf{KEM.Gen}^*$, $\mathsf{KEM.Enc}^*$, and $\mathsf{KEM.Dec}^*$ are the same as $\mathsf{KEM.Gen}$, $\mathsf{KEM.Enc}$, and $\mathsf{KEM.Dec}$, respectively. The algorithms $\mathsf{DEM.Enc}^*$, $\mathsf{DEM.Dec}^*$ are defined as follows.

$\mathsf{DEM.Enc}^*$**:** On input a symmetric key K and a message $M \in \{0,1\}^m$, choose $r' \in \{0,1\}^t$ uniformly at random. Output the ciphertext

$$d = (\mathsf{Ext}(K, r') \oplus M, r').$$

$\mathsf{DEM.Dec}^*$ On input a symmetric key K and a ciphertext $d = (d_1, d_2)$, output the message

$$M = \mathsf{Ext}(K, d_2) \oplus d_1.$$

Proof. We define the experiments $\mathsf{Expt}^{\mathsf{RandLeak}^*}_{\Pi^*,A}(b)$ and $\mathsf{Expt}_{\Pi^*,A}(b)$ for $b \in \{0,1\}$ as follows.

$\mathsf{Expt}^{\mathsf{RandLeak}^*}_{\Pi^*,A}(b)$:

1. Choose pk^* from PK^*, where PK^* is the distribution defined in Definition 8.
2. Choose $r \in R$ uniformly at random, where R is the domain of the randomness used in $\mathsf{KEM.Enc}^*$.
3. $(c^*, K^*) \leftarrow \mathsf{KEM.Enc}^*(pk^*, r)$.
4. $(m_0, m_1, st_1) \leftarrow A_1^{\mathsf{Leak}(r)}(pk^*)$ such that $|m_0| = |m_1|$.
5. Choose $r' \in R'$ uniformly at random, where R' is the domain of the randomness used in $\mathsf{DEM.Enc}^*$.
6. $d \leftarrow \mathsf{DEM.Enc}^*(m_b, K^*, r')$.
7. $b' \leftarrow A_2^{\mathsf{Leak}'(r')}(c^*, d, st_1)$.
8. Output b'.

$\mathsf{Expt}_{\Pi^*,A}(b)$:

1. Choose pk^* from PK^*.
2. Choose $r \in R$ uniformly at random.
3. $(c^*, K^*) \leftarrow \mathsf{KEM.Enc}^*(pk^*, r)$.
4. $(m_0, m_1, st_1) \leftarrow A_1^{\mathsf{Leak}(r)}(pk^*)$ such that $|m_0| = |m_1|$.
5. Choose $r' \in R'$ uniformly at random.
6. Choose $r^* \leftarrow U_m$, and set $d = (r^* \oplus m_b, r')$.
7. $b' \leftarrow A_2^{\mathsf{Leak}'(r')}(c, d, st_1)$.
8. Output b'.

Using the triangle inequality, for any adversary A it holds that

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{RandLeak}}_{\Pi^*,A}(n)\\
&= \left|\Pr[\mathsf{Expt}^{\mathsf{RandLeak}}_{\Pi^*,A}(0)=1] - \Pr[\mathsf{Expt}^{\mathsf{RandLeak}}_{\Pi^*,A}(1)=1]\right|\\
&\le \left|\Pr[\mathsf{Expt}^{\mathsf{RandLeak}}_{\Pi^*,A}(0)=1] - \Pr[\mathsf{Expt}^{\mathsf{RandLeak}^*}_{\Pi^*,A}(0)=1]\right| && (1)\\
&\quad + \left|\Pr[\mathsf{Expt}^{\mathsf{RandLeak}^*}_{\Pi^*,A}(0)=1] - \Pr[\mathsf{Expt}_{\Pi^*,A}(0)=1]\right| && (2)\\
&\quad + \left|\Pr[\mathsf{Expt}_{\Pi^*,A}(0)=1] - \Pr[\mathsf{Expt}_{\Pi^*,A}(1)=1]\right| && (3)\\
&\quad + \left|\Pr[\mathsf{Expt}_{\Pi^*,A}(1)=1] - \Pr[\mathsf{Expt}^{\mathsf{RandLeak}^*}_{\Pi^*,A}(1)=1]\right| && (4)\\
&\quad + \left|\Pr[\mathsf{Expt}^{\mathsf{RandLeak}^*}_{\Pi^*,A}(1)=1] - \Pr[\mathsf{Expt}^{\mathsf{RandLeak}}_{\Pi^*,A}(1)=1]\right|. && (5)
\end{aligned}$$

We first show an upper bound on the terms (1) and (5). Note that, the difference between $\mathsf{Expt}^{\mathsf{RandLeak}}_{\Pi^*,A}(b)$ and $\mathsf{Expt}^{\mathsf{RandLeak}^*}_{\Pi^*,A}(b)$ is only the distribution of choosing public keys. Since the distribution PK^* and $\{pk \,|\, (pk, sk) \leftarrow \mathsf{KEM.Gen}(1^n)\}$ are computationally indistinguishable, there is some negligible function $\mathsf{AdvComp}(n)$ that is an upper bound on both (1) and (5).

Second, we show an upper bound on (2) and (4). The difference between $\mathsf{Expt}_{\Pi^*,A}(b)$ and $\mathsf{Expt}^{\mathsf{RandLeak}^*}_{\Pi^*,A}(b)$ is only the mask string for m_b, which is $\mathsf{Ext}(K^*, r')$ and r^*, respectively. From the property of the KEM scheme, there is a distribution K' such that $\tilde{H}_\infty(K'|pk^*, c^*, f(pk^*, r)) \ge \kappa(n) - \lambda(n)$ and $\Delta((pk^*, c^*, K^*, f(pk^*, r)), (pk^*, c^*, K', f(pk^*, r))) \le \epsilon_2(n)$. Since r' is chosen from U_t and Ext is an average-case $(\kappa(n) - \lambda(n), \epsilon_1(n))$-strong extractor, we have that

$$\begin{aligned}
\Delta((&\mathsf{Ext}(K', r'), r', pk^*, c^*, f(pk^*, r)),\\
&(r^*, r', pk^*, c^*, f(pk^*, r))) \le \epsilon_1(n),
\end{aligned}$$

and thus

$$\begin{aligned}
\Delta((&\mathsf{Ext}(K^*, r'), r', pk^*, c^*, f(pk^*, r)),\\
&(r^*, r', pk^*, c^*, f(pk^*, r))) \le \epsilon_1(n) + \epsilon_2(n).
\end{aligned}$$

Therefore, $\epsilon_1(n) + \epsilon_2(n)$ is an upper bound on both (2) and (4).

Finally, we show the term (3) is equal to zero. The difference between $\mathsf{Expt}_{\Pi^*,A}(0)$ and $\mathsf{Expt}_{\Pi^*,A}(1)$ is the message m_b for $b \in \{0,1\}$. Since m_b is masked by a uniformly random string r^*, the experiments $\mathsf{Expt}_{\Pi^*,A}(0)$ and $\mathsf{Expt}_{\Pi^*,A}(1)$ are the same. Thus, (3) is equal to zero.

Therefore, we have $\mathsf{Adv}^{\mathsf{RandLeak}}_{\Pi^*,A}(n) \le 2(\mathsf{AdvComp}(n) + \epsilon_1(n) + \epsilon_2(n))$, which is negligible in n. □

6. The Construction of Entropically-Secure KEM

In this section, we provide a construction of entropically secure KEM from hash proof systems. Our construction is based on the KEM scheme of Naor and Segev [15], which is entropically secure against "secret-key" leakage attack. To achieve the "randomness" leakage resilience, we exchange the roles of "secret key" and "randomness" in the Naor-Segev (NS) scheme. Specifically, in our construction, the KEM encryption of the NS scheme is performed in the key generation algorithm, the key generation of the NS scheme is performed in the encryption algorithm. That is, a ciphertext of the NS scheme is used as a public key, and a public key of the NS scheme is used as a ciphertext in our construction. In general, such an exchange does not work since (1) a ciphertext may be generated depending on the public key, and (2) the encapsulated key must be decapsulated from the secret key and the ciphertext. Both of the two points can be circumvented by hash proof systems. First, a ciphertext is just a random sample from the valid ciphertext space, and is independent of public keys. Second, in hash proof systems, there are two algorithms Pub and $\Lambda_{(\cdot)}$ such that $\mathsf{Pub}(pk, c, w) = \Lambda_{sk}(c)$, where $c \in \mathcal{V}$ is a valid ciphertext and w is the corresponding witness. Thus, exchanging the roles of secret key and randomness correctly works for hash proof systems.

Construction 3: Let $\mathsf{HPS} = (\mathsf{Param}, \mathsf{Pub}, \mathsf{Priv})$ be a hash proof system with $\delta(n)$-smoothness. The KEM scheme $(\mathsf{KEM.Gen}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec})$ is defined as follows.

$\mathsf{KEM.Gen}$: On input a security parameter 1^n, $F = (\mathcal{SK}, \mathcal{PK}, C, \mathcal{V}, \mathcal{K}, \Lambda, \mu) \leftarrow \mathsf{Param}(1^n)$, and choose $c \in \mathcal{V}$ uniformly at random together with the corresponding witness w. Output a public key $PK = (F, c)$ and a secret key $SK = w$.

$\mathsf{KEM.Enc}$: On input a public key $pk = (F, c)$, choose $sk \in \mathcal{SK}$ uniformly at random, and compute $pk = \mu(sk)$ and $K = \Lambda_{sk}(c)$. Output pk as the ciphertext, and K as the symmetric key.

$\mathsf{KEM.Dec}$: On input a public key $PK = (F, c)$, a secret key $SK = w$ and a ciphertext pk, output the symmetric key $\mathsf{Pub}(pk, c, w)$.

The correctness of the above scheme follows from the property of hash proof systems since for any public key (F, c), ciphertext pk, and a secret key w, it holds that $\mathsf{Pub}(pk, c, w) = \Lambda_{sk}(c)$.

Theorem 3: The KEM scheme defined in Construction 3 is $(\log|\mathcal{K}| - \lambda(n), \delta(n))$-entropically secure against $\lambda(n)$-randomness-leakage attack.

Proof. We need to show that (1) there exists a distribution PK^* such that PK^* is computationally indistinguishable from the distribution $\{pk \,|\, pk \leftarrow \mathsf{KEM.Enc}(1^n)\}$, and that (2) there is a distribution K' such that

$$\tilde{H}_\infty(K'|pk^*, c^*, f(pk^*, r)) \geq \log|\mathcal{K}| - \lambda(n)$$

and

$$\Delta((pk^*, c^*, K^*, f(pk^*, r)), (pk^*, c^*, K', f(pk^*, r))) \leq \delta(n),$$

where $pk^* \leftarrow PK^*$, $(c^*, K^*) \leftarrow \mathsf{KEM.Enc}(pk^*, r)$.

Regarding (1), we define PK^* as follows.

PK^***:** On input 1^n, $F = (\mathcal{SK}, \mathcal{PK}, C, \mathcal{V}, \mathcal{K}, \Lambda, \mu) \leftarrow \mathsf{Param}(1^n)$, and choose $c' \in C \setminus \mathcal{V}$ uniformly at random, and output (F, c')

It immediately follows from the hardness of subset membership problem that PK^* is computationally indistinguishable from the distribution $\{pk \,|\, pk \leftarrow \mathsf{KEM.Enc}(1^n)\}$.

Regarding (2), let K' be a uniform distribution over $\mathcal{K}$. Then, it follows from Lemma 1 that $\tilde{H}_\infty(K'|pk^*, c^*, f(pk^*, r)) = \tilde{H}_\infty(K'|F, c', \mu(sk), f(pk^*, sk)) \geq \tilde{H}_\infty(K'|F, c', \mu(sk)) - \lambda(n) = \log|\mathcal{K}| - \lambda(n)$. The $\delta(n)$-smoothness of the hash proof system implies that

$$\begin{aligned}&\Delta((pk^*, c^*, K^*), (pk^*, c^*, K'))\\ &= \Delta((F, c', \mu(sk), \Lambda_{sk}(c')), (F, c', \mu(sk), K'))\\ &\leq \delta(n).\end{aligned}$$

We need to consider the statistical distance given any leakage of $f(pk^*, r)(= f(F, c', sk))$. Note that, as also discussed in [15] (the proof of Claim 4.2 of the full version), the distribution of $f(F, c', sk)$ is fully determined by $F, c', \mu(sk)$, and $\Lambda_{sk}(c')$. Let f' be a function such that $f'(F, c', \mu(sk), \Lambda_{sk}(c'))$ is identically distributed to $f(F, c', sk)$. Then, by using the fact that applying the same function to two distributions cannot increase their statistical distance, we have that

$$\begin{aligned}&\Delta((F, c', \mu(sk), \Lambda_{sk}(c'), f'(F, c', \mu(sk), \Lambda_{sk}(c')),\\ &\quad (F, c', \mu(sk), K', f'(F, c', \mu(sk), \Lambda_{sk}(c'))))) \leq \delta(n).\end{aligned}$$

Therefore, the statement follows. □

A DDH-based Instantiation

We give an instantiation of Construction 3 based on the decisional Diffie-Hellman (DDH)-based hash proof system.

Construction 4: Let G be a group of prime order p, and $\lambda(n)$ a leakage parameter. Then, the KEM scheme $(\mathsf{KEM.Gen}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec})$ is defined as follows.

$\mathsf{KEM.Gen}$: On input a security parameter 1^n, choose $x_1 \in \mathbb{Z}_p$ and $g_1, g_2 \in G$ uniformly at random. Output a pair of keys (pk, sk) as

$$pk = (g_1, g_2, g_1^{x_1}, g_2^{x_1}), \qquad sk = x_1.$$

$\mathsf{KEM.Enc}$: On input a public key $pk = (g_1, g_2, pk_1, pk_2)$, choose $r_1, r_2 \in \mathbb{Z}_p$ uniformly at random, and output the ciphertext c and the symmetric key K as

$$c = g_1^{r_1} g_2^{r_2}, \qquad K = (pk_1)^{r_1}(pk_2)^{r_2}.$$

$\mathsf{KEM.Dec}$: On inputs a secret key sk and a ciphertext c, output the symmetric key as

$$K = c^{sk}.$$

The correctness of the scheme immediately follows since if $c = g_1^{r_1} g_2^{r_2}$, $pk_1 = g_1^{x_1}$ and $pk_2 = g_2^{x_1}$, then $K = (pk_1)^{r_1}(pk_2)^{r_2} = (g_1^{x_1})^{r_1}(g_2^{x_1})^{r_2} = (g_1^{r_1} g_2^{r_2})^{x_1} = c^{x_1}$.

Since the DDH-based hash proof system has 0-smoothness, the above instantiation of KEM is $(\log p, 0)$-entropically secure under the DDH assumption.

See [15] for other constructions of hash proof systems.

7. Conclusions

In this paper, we have studied the security of public-key encryption against the leakage of randomness for encryption. In contrast to the secret key leakage, it is impossible to achieve the security against leakage if the leakage can depend on the public key. To circumvent this impossibility result, we have considered the leakage model for the KEM/DEM framework such that the leakage of randomness for KEM and DEM occurs independently. Then, we have constructed a KEM/DEM scheme secure in this model from hash proof systems. The construction is based on the secret-key leakage-resilient KEM/DEM scheme of Naor and Segev [15]. We have managed to convert their "secret-key" leakage-resilient scheme to a "randomness" leakage-resilient scheme.

Acknowledgments

This work was supported in part by NTT Information Sharing Platform Laboratories, the Telecommunications Advancement Foundation, and Grant-in-Aid for Scientific Research.

References

[1] A. Akavia, S. Goldwasser, and V. Vaikuntanathan, "Simultaneous hardcore bits and cryptography against memory attacks," in Omer Reingold, ed., Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, Lect. Notes Comput. Sci., vol.5444, pp.474–495, Springer-Verlag, New York, USA, March 2009.

[2] J. Alwen, Y. Dodis, and D. Wichs, "Leakage-resilient public-key cryptography in the bounded-retrieval model," in Advances in Cryptology — CRYPTO 2009, Lect. Notes Comput. Sci., pp.36–54, Springer, Santa Barbara, California, USA, Aug. 2009.

[3] J. Alwen, Y. Dodis, and D. Wichs, "Survey: Leakage resilience and the bounded retrieval model," in Kaoru Kurosawa, ed., ICITS, Lect. Notes Comput. Sci., vol.5973, pp.1–18, Springer, 2009.

[4] M. Bellare, Z. Brakerski, M. Naor, T. Ristenpart, G. Segev, H. Schacham, and S. Yilek, "Hedged public-key encryption: How to protect against bad randomness," in Mitsuru Matsui, ed., Advances in Cryptology — ASIACRYPT 2009, Lect. Notes Comput. Sci., vol.5912, pp.232–249, Springer, Tokyo, Japan, Dec. 2009.

[5] R. Canneti, Y. Dodis, S. Halevi, E. Kushilevitz, and A. Sahai, "Exposure-resilient functions and all-or-nothing transforms," in Bart Preneel, ed., Advances in Cryptology — EUROCRYPT 2000, Lect. Notes Comput. Sci., vol.1807, pp.453–469, Springer-Verlag, Bruges, Belgium, May 2000.

[6] R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption," in L. Knudsen, ed., EUROCRYPT, Lect. Notes Comput. Sci., vol.2332, pp.45–64, Springer-Verlag, Amsterdam, The Netherlands, April 2002.

[7] F. Davì, S. Dziembowski, and D. Venturi, "Leakage-resilient storage," in J.A. Garay and R. De Prisco, eds., SCN, Lect. Notes Comput. Sci., vol.6280, pp.121–137, Springer, 2010.

[8] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," SIAM J. Comput., vol.38, no.1, pp.97–139, 2008.

[9] Y. Dodis, Y.T. Kalai, and S. Lovett, "On cryptography with auxiliary input," in Michael Mitzenmacher, ed., Proc. 41st Annual ACM Symposium on Theory of Computing, STOC 2009, pp.621–630, Bethesda, MD, USA, ACM, 2009.

[10] S. Dziembowski and K. Pietrzak, "Leakage-resilient cryptography," in 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008), IEEE Computer Society, pp.293–302, Philadelphia, PA, USA, Oct. 2008.

[11] J.A. Halderman, S.D. Schoen, N. Heninger, W. Clarkson, W. Paul, J.A. Calandrino, A.J. Feldman, J. Appelbaum, and E.W. Felten, "Lest we remember: Cold boot attacks on encryption keys," in USENIX Security Symposium, pp.45–60, 2008.

[12] S. Halevi and H. Lin, "After-the-fact leakage in public-key encryption," in Yuval Ishai, ed., TCC, Lect. Notes Comput. Sci., vol.6597, pp.107–124, Springer, 2011.

[13] S. Kamara and J. Katz, "How to encrypt with a malicious random number generator," in K. Nyberg, ed., FSE, Lect. Notes Comput. Sci., vol.5086, pp.303–315, Springer-Verlag, Lausanne, Switzerland, Feb. 2008.

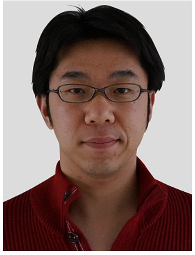
[14] F.-H. Liu and A. Lysyanskaya, "Tamper and leakage resilience in the split-state model," in R. Safavi-Naini and R. Canetti, eds., CRYPTO, Lect. Notes Comput. Sci., vol.7417, pp.517–532, Springer, 2012.

[15] M. Naor and G. Segev, "Public-key cryptosystems resilient to key leakage," in S. Halevi, ed., CRYPTO, Lect. Notes Comput. Sci., vol.5677, pp.18–35, Springer, 2009. Full version is available at http://eprint.iacr.org/2009/105.pdf

[16] K. Pietrzak, "Provable security for physical cryptography," in Western European Workshop on Research in Cryptology — WEWoRC 2009, 2009.

Hitoshi Namiki received his B.S. and M.S. degrees from Tokyo Institute of Technology in 2008 and 2010, respectively. Currently he works at Ricoh Co. Ltd.

Keisuke Tanaka is Associate Professor of Department of Mathematical and Computing Sciences at Tokyo Institute of Technology. He received his B.S. from Yamanashi University in 1992 and his M.S. and Ph.D. from Japan Advanced Institute of Science and Technology in 1994 and 1997, respectively. For each degree, he majored in computer science. Before joining Tokyo Institute of Technology, he was Research Engineer at NTT Information Platform Labs.



Kenji Yasunaga received his B.E. degree in information and computer sciences in 2003, and his M.S. and Ph.D. degrees in information science and technology in 2005 and 2008, from Osaka University, Japan. He was a Postdoctoral Fellow at Kwansei Gakuin University in 2008, was an Assistant Professor at Tokyo Institute of Technology from 2008 to 2011, and was a Researcher at Institute of Systems, Information Technologies and Nanotechnologies (ISIT) from 2011 to 2012. He is currently an Assistant Professor at Kanazawa University. His research interests are in Coding Theory, Cryptography, and Computational Complexity.