Randomness Leakage in the KEM/DEM Framework

Hitoshi Namiki[1], Keisuke Tanaka[2], and Kenji Yasunaga[2]

[1] Ricoh Co., Ltd., Tokyo
[2] Tokyo Institute of Technology

Abstract. 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 under the decisional Diffie-Hellman assumption.

1 Introduction

Recently, many studies have been devoted to construct encryption schemes that are secure even if some of the secret information leaks [4, 7, 1, 11, 2, 3]. In [4, 7, 1, 11, 2], they mainly considered the case of leaking a secret key. In particular, studies in [1, 11, 2] 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 [8]. Naor and Segev [11] 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. [3] 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, 11]. 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 [11]. However, for the a posteriori randomness-leakage attack, we show that no public-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 [11]. The results are summarized in Table 1, in which we compare the results of randomness leakage with that of key leakage.

Table 1. Comparison between Key Leakage and Randomness Leakage.

Timing of leakage	Key leakage	Randomness leakage
Before receiving the public key	IND-CPA PKE [11]	IND-CPA PKE
After receiving the public key	Hash Proof Systems [11]	Impossible
After receiving the ciphertext	Impossible [1]	Impossible

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 selected 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 is due to some technical reason. Hence, removing the second condition can be considered future work of this study. 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 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 for KEM. We call it *the entropic security against randomness-leakage attack*. A KEM scheme is entropically secure against randomness-leakage attack 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 has high entropy even if the randomness leakage occurs. Then, we provide a generic construction of a public-key encryption scheme that is resilient to randomness leakage from any KEM scheme that is entropically secure against randomness-leakage attack.

Also we construct a KEM scheme that is entropically secure against randomness-leakage attack under the decisional Diffie-Hellman (DDH) assumption. The scheme is a simple variant of the ElGamal KEM scheme.

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.

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
[11]	Secret key	HPS	After receiving the public key
[3]	Randomness	Lossy TDF	Before receiving the public key
This work	Randomness	DDH	After receiving the public key

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 [11] 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. Recently, Halevi and Lin [9] have studied a realizable security of public-key encryption schemes in which the leakage occurs after the adversary receives the ciphertext.

There are several studies related to the leakage of the randomness used in the encryption algorithm. Bellare et al. [3] 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 [10] 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 [7] considered the key leakage under the assumption that only the computation leaks information, and constructed leakage-resilient stream-ciphers. Dodis, Tauman Kalai, and Lovett [6] 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. We say an algorithm is PPT if it runs by a probabilistic polynomial-time Turing machine.

2.1 The Decisional Diffie-Hellman Assumption

Let $\mathcal{G}(1^n)$ be a group sampling algorithm which, on input 1^n, outputs a tuple of $\mathbb{G} = (p, G, g)$ where p is a prime, G is a group of order p, and g is a generator of G.

The decisional Diffie-Hellman (DDH) assumption is that the ensembles $\{(\mathbb{G}, g_1, g_2, g_1^r, g_2^r)\}_{n\in\mathbb{N}}$ and $\{(\mathbb{G}, g_1, g_2, g_1^{r_1}, g_2^{r_2})\}_{n\in\mathbb{N}}$ are computationally indistinguishable, where $\mathbb{G} \leftarrow \mathcal{G}(1^n)$, the elements g_1, g_2 are randomly selected generator of G, and $r, r_1, r_2 \in \mathbb{Z}_p$ are chosen independently and uniformly at random.

2.2 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 [5], 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, $\tilde{H}_\infty(X|(Y,Z)) \geq H_\infty(X|Z) - r$.*

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 [5].

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

2.3 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 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*

$\mathsf{KEM.Gen}$: *On input a security parameter* 1^n*, output a pair of keys* (pk, sk)*.*
$\mathsf{KEM.Enc}$: *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*.*
$\mathsf{KEM.Dec}$: *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*

$\mathsf{DEM.Gen}$: *On input a security parameter* 1^n*, output a symmetric key* K*.*
$\mathsf{DEM.Enc}$: *On input a symmetric key* K*, a message* m*, and a random string* r*, output a ciphertext* c*.*
$\mathsf{DEM.Dec}$: *On input a symmetric key* K *and a ciphertext* c*, output a message* m*.*

Note that we define $\mathsf{DEM.Enc}$ as a randomized algorithm, while it is usually deterministic. We provide a KEM/DEM construction in Section 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 $(\mathsf{KEM.Gen}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec}, \mathsf{DEM.Enc}, \mathsf{DEM.Dec})$.

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, and $\ell(n)$ the length of the randomness used in the encryption algorithm 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 4. *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 scheme 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 5 *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* $\mathsf{Ext} : \{0,1\}^{k(n)} \times \{0,1\}^{t(n)} \to \{0,1\}^{m(n)}$ *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 $\mathsf{Adv}^{\mathsf{IND-CPA}}_{\Pi,A'}(n)$ 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 4 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 $\mathsf{Expt}^{\mathsf{IND-CPA}}_{\Pi,A}(b)$, 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 4. 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*

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

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 KEM/DEM-based public-key encryption schemes. As discussed in Section 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 $\mathsf{KEM.Enc}$ and $\mathsf{DEM.Enc}$ occur independently. Also when the adversary chooses two challenge messages, she is not allowed to access to the leakage information of random bits in $\mathsf{DEM.Enc}$.

We describe the formal definition of the randomness-leakage attack in KEM/DEM. The randomness-leakage oracle for $\mathsf{KEM.Enc}$, 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 $\mathsf{KEM.Enc}$ and r is the random bits generated in $\mathsf{KEM.Enc}$. We restrict the function f to be efficiently computable. The leakage oracle for $\mathsf{DEM.Enc}$, 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 $\mathsf{DEM.Enc}$ and r' is the random bits generated in $\mathsf{DEM.Enc}$. 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 $\mathsf{DEM.Enc}$. 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* $\Pi = (\mathsf{KEM.Gen}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec}, \mathsf{DEM.Enc}, \mathsf{DEM.Dec})$ *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,*

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

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. $(m_0, m_1, st_1) \leftarrow A_1^{\mathsf{Leak}(r)}(pk, c)$ *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. $b' \leftarrow A_2^{\mathsf{Leak}'(r')}(d, st_1)$.
8. *Output b'.*

Note that the ciphertext c generated by $\mathsf{KEM.Enc}$ is given to the adversary before she submits the challenge messages m_0 and m_1. This means the above definition captures a stronger security than the standard KEM/DEM framework. The randomness r' in $\mathsf{DEM.Enc}$ leaks only after the adversary chose the challenge messages.

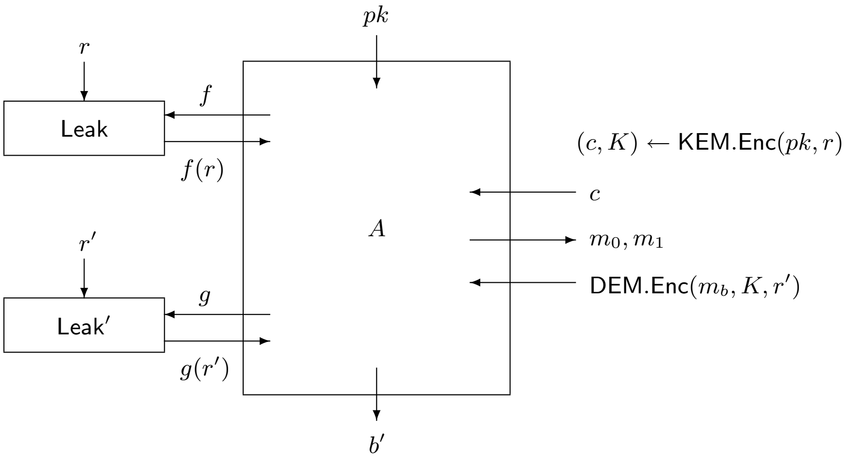
pk
r
Leak
f
$f(r)$
$(c, K) \leftarrow \mathsf{KEM.Enc}(pk, r)$
c
A
m_0, m_1
r'
$\mathsf{DEM.Enc}(m_b, K, r')$
g
Leak$'$
$g(r')$
b'

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

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 encryption algorithm of KEM.

Given $pk^* \in PK^*, c^*$, and $f(r)$, we require that K^* 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* $(\mathsf{KEM.Gen}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec})$ *is* $\kappa(n)$-entropically secure against $\lambda(n)$-randomness-leakage attack *if there exists an efficiently samplable distribution* PK^* *that is computationally indistinguishable from the distribution* $\{pk \mid (pk, sk) \leftarrow \mathsf{KEM.Gen}(1^n)\}$*, and*

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

where $pk^* \leftarrow PK^*$*,* $(c^*, K^*) \leftarrow \mathsf{KEM.Enc}(pk^*, r)$*, 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* $\mathsf{Ext} : G \times \{0,1\}^t \rightarrow \{0,1\}^m$ *be an average-case* $(\kappa(n), \epsilon(n))$*-strong extractor for some negligible function* $\epsilon(n)$*. Let* $(\mathsf{KEM.Gen}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec})$ *be a KEM scheme that is* $\kappa(n)$*-entropically secure against* $\lambda(n)$*-randomness-leakage attack. Then, the following scheme* $\Pi^* = (\mathsf{KEM.Gen}^*, \mathsf{KEM.Enc}^*, \mathsf{KEM.Dec}^*, \mathsf{DEM.Enc}^*, \mathsf{DEM.Dec}^*)$ *is a public-key encryption scheme that is IND-CPA secure against* $\lambda(n)$*-randomness-leakage attack.*

Construction 9 *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 (Proof of Theorem 2). 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. $(m_0, m_1, st_1) \leftarrow A_1^{\mathsf{Leak}(r)}(pk^*, c^*)$ 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. $b' \leftarrow A_2^{\mathsf{Leak}'(r')}(d, st_1)$.
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. $(m_0, m_1, st_1) \leftarrow A_1^{\mathsf{Leak}(r)}(pk^*, c^*)$ 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. $b' \leftarrow A_2^{\mathsf{Leak}'(r')}(d, st_1)$.
8. Output b'.

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

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

We first show an upper bound on the terms (1) and (5). Note that, the difference between $\mathsf{Expt}_{\Pi^*,A}^{\mathsf{RandLeak}}(b)$ and $\mathsf{Expt}_{\Pi^*,A}^{\mathsf{RandLeak}^*}(b)$ 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 $\mathsf{Expt}_{\Pi^*,A}^{\mathsf{RandLeak}^*}(b)$ is only the mask string for m_b, which is $\mathsf{Ext}(K^*, r')$ and r^*, respectively. From the assumption of Theorem 2, we have that $\widetilde{H}_\infty(K^*|pk^*, c^*, f(r)) \ge \kappa(n)$. Since r' is chosen from U_t and Ext is an average-case $(\kappa(n), \epsilon(n))$-strong extractor, the adversary can distinguish the distribution between $(\mathsf{Ext}(K^*, r'), r', pk^*, c^*, f(r))$ and $(r^*, r', pk^*, c^*, f(r))$ with probability at most $\epsilon(n)$. Thus, $\epsilon(n)$ 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 that $\mathsf{Adv}^{\mathsf{RandLeak}}_{\Pi^*,A}(n) \leq 2(\mathsf{AdvComp}(n) + \epsilon(n))$, which is negligible in n.

6 The Construction of Entropically-Secure KEM

In this section, we provide a construction of entropically secure KEM based on the DDH assumption.

Construction 10 *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 $K = (pk_1)^{r_1}(pk_2)^{r_2} = (g_1^{x_1})^{r_1}(g_2^{x_1})^{r_2} = (g_1^{r_1} g_2^{r_2})^{x_1} = c^{x_1}$.

Next, we show the security of the scheme.

Theorem 3. *The KEM scheme defined in Construction 10 is $(\log p - \lambda(n))$-entropically secure against $\lambda(n)$-randomness-leakage attack under the DDH assumption.*

Proof. We need to show that there exists a distribution PK^* such that PK^* is computationally indistinguishable from the distribution $\{pk \,|\, pk \leftarrow \mathsf{KEM.Enc}(1^n)\}$, and that $\tilde{H}_\infty(K^*|pk^*, c^*, f(r^*)) \geq \log p - \lambda(n)$, where $pk^* \leftarrow PK^*$, $(c^*, K^*) \leftarrow \mathsf{KEM.Enc}(pk^*, r)$, and f is an arbitrary efficiently-computable function whose output length is at most $\lambda(n)$.

We define PK^* as follows.

PK^***:** Choose $x_1, x_2 \in \mathbb{Z}_p$ with $x_1 \neq x_2$ and $g_1, g_2 \in G$ uniformly at random. Then output $pk^* = (g_1, g_2, g_1^{x_1}, g_2^{x_2})$.

It follows from the DDH assumption that the distribution PK^* and the distribution $\{pk \,|\, pk \leftarrow \mathsf{KEM.Enc}(1^n)\} = \{(g_1, g_2, g_1^{x_1}, g_2^{x_1}) \,|\, x_1 \in \mathbb{Z}_p, g_1, g_2 \in G\}$ are computationally indistinguishable.

Next, we show that $\widetilde{H}_\infty(K^*|pk^*, c^*, f(r^*)) \geq \log p - \lambda(n)$. If pk^* is chosen from PK^*, then the distribution $\{K^* \,|\, (c^*, K^*) \leftarrow \mathsf{KEM.Enc}(pk^*, r)\}$ is equal to the following distribution $\mathcal{K}^*$.

$\mathcal{K}^*$: Choose $x_1, x_2, r_1, r_2 \in \mathbb{Z}_p$ with $x_1 \neq x_2$ and $g_1, g_2 \in G$ uniformly at random, and compute $c^* = g_1^{r_1} g_2^{r_2}$ and $K^* = (g_1^{x_1})^{r_1}(g_2^{x_2})^{r_2}$. Then output K^*.

We show that, given pk^* and c^*, the distribution $\mathcal{K}^*$ is the uniform distribution on G. To prove this fact, we show that, given pk^* and c^*, for any $K^* \in G$, there is a unique pair $(r_1, r_2) \in \mathbb{Z}_p \times \mathbb{Z}_p$ that satisfies $g_1^{r_1} g_2^{r_2} = c^*$ and $(g_1^{x_1})^{r_1}(g_2^{x_2})^{r_2} = K^*$. We can write $g_2 = g_1^{\alpha}, c^* = g_1^{y_1}$, and $K^* = g_1^{y_2}$ for some $\alpha, y_1, y_2 \in \mathbb{Z}_p$. Then it holds that $c^* = g_1^{y_1} = g_1^{r_1 + \alpha r_2}$ and that $K^* = g_1^{y_2} = g_1^{x_1 r_1 + \alpha x_2 r_2}$. Hence, we have two equations $y_1 = r_1 + \alpha r_2$ and $y_2 = x_1 r_1 + \alpha x_2 r_2$. Since $x_1 \neq x_2$ and $\alpha \neq 0$ (otherwise, g_2 is not a generator of G), there is a unique solution (r_1, r_2) of these equations.

From the above, we have that $H_\infty(K^*|pk^*, c^*) = \log p$. Then it follows from Lemma 1 that $\widetilde{H}_\infty(K^*|pk^*, c^*, f(r)) \geq H_\infty(K^*|pk^*, c^*) - \lambda(n) = \log p - \lambda(n)$.

In summary, from Theorems 2 and 3, we can construct a public-key encryption scheme secure against randomness-leakage attack.

Theorem 4. *Let G be a group of prime order p, and $\mathsf{Ext} : G \times \{0,1\}^t \rightarrow \{0,1\}^m$ an average-case $(\log p - \lambda(n), \epsilon(n))$-strong extractor for some negligible function $\epsilon(n)$. Then, the following public-key encryption scheme $\Pi =$ $(\mathsf{KEM.Gen}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec}, \mathsf{DEM.Enc}, \mathsf{DEM.Dec})$ is IND-CPA secure against $\lambda(n)$-randomness-leakage attack under the DDH assumption.*

$\mathsf{KEM.Gen}$: *On input a security parameter 1^n, choose $x_1 \in \mathbb{Z}_p$ and $g_1, g_2 \in G$ uniformly at random, and output the public key $pk = (g_1, g_2, g_1^{x_1}, g_2^{x_1})$ and the secret key $sk = x_1$.*

$\mathsf{KEM.Enc}$: *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 $c = g_1^{r_1} g_2^{r_2}$ and the symmetric key $K = (pk_1)^{r_1}(pk_2)^{r_2}$.*

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

$\mathsf{DEM.Enc}$: *On input a symmetric key K and a message $M \in \{0,1\}^m$, choose $r' \in \{0,1\}^t$ uniformly at random, and 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 $M = \mathsf{Ext}(K, d_2) \oplus d_1$.*

Acknowledgments

This work was supported in part by NTT Information Sharing Platform Laboratories 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 O. Reingold, editor, *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009*, volume 5444 of *Lecture Notes in Computer Science*, pages 474–495, New York, USA, March 2009. Springer-Verlag.
2. J. Alwen, Y. Dodis, and D. Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *Advances in Cryptology – CRYPTO 2009*, Lecture Notes in Computer Science, pages 36–54, Santa Barbara, California, USA, August 2009. Springer.
3. 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 M. Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 232–249, Tokyo, Japan, December 2009. Springer.
4. R. Canneti, Y. Dodis, S. Halevi, E. Kushilevitz, and A. Sahai. Exposure-resilient functions and all-or-nothing transforms. In B. Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 453–469, Bruges, Belgium, May 2000. Springer-Verlag.
5. 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.*, 38(1):97–139, 2008.
6. Y. Dodis, Y. Tauman Kalai, and S. Lovett. On cryptography with auxiliary input. In M. Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 621–630. ACM, 2009.
7. S. Dziembowski and K. Pietrzak. Leakage-resilient cryptography. In *49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*, pages 293–302, Philadelphia, PA, USA, October 2008. IEEE Computer Society.
8. 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*, pages 45–60, 2008.
9. S. Halevi and H. Lin. After-the-fact leakage in public-key encryption. In Y. Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 107–124. Springer, 2011.
10. S. Kamara and J. Katz. How to encrypt with a malicious random number generator. In K. Nyberg, editor, *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 303–315, Lausanne, Switzerland, February 2008. Springer-Verlag.
11. M. Naor and G. Segev. Public-key cryptosystems resilient to key leakage. In *Advances in Cryptology – CRYPTO 2009*, Lecture Notes in Computer Science, pages 18–35, Santa Barbara, California, USA, August 2009. Springer.