12-2016

Lossy key encapsulation mechanism and its applications

Yamin LIU

Xianhui LU

Bao LI

Haiyang XUE
Singapore Management University, haiyangxue@smu.edu.sg

Lossy Key Encapsulation Mechanism and Its Applications

Yamin Liu[1,2(✉)], Xianhui Lu[1,2,3], Bao Li[1,2,3], and Haiyang Xue[1,2]

[1] Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, Beijing, China
[2] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{liuyamin,luxianhui,libao,xuehaiyang}@iie.ac.cn
[3] University of Chinese Academy of Sciences, Beijing, China

Abstract. We introduce a new notion, lossy key encapsulation mechanism (lossy KEM), which enhances the notion of key encapsulation mechanism with lossiness, and can be more efficient than lossy trapdoor functions. We show that lossy KEM can be constructed from lossy trapdoor functions, lossy trapdoor relations, and entropic projective hashing. Using lossy KEM as a building block, several previous constructions of lossy encryption and deterministic public key encryption can be generalized and improved in efficiency.

Keywords: Lossy key encapsulation mechanism · Lossy encryption · Deterministic public key encryption

1 Introduction

Lossy Primitives. Lossy primitives became important building blocks of various cryptosystems in the last decades. The first lossy primitive, lossy trapdoor function (LTDF), was introduced by Peikert and Waters in 2008 [20]. LTDF is useful in building plenty of cryptographic schemes, e.g., oblivious transfer, collision-resistant hash, leakage-resilient encryption, chosen ciphertext-secure encryption, and deterministic public-key encryption (DPKE). LTDF can be constructed from various number-theoretic assumptions and lattice-based assumptions [15,24], and from dual projective hashing [23].

In 2009, Bellare *et al.* introduced lossy encryption [6], which implies indistinguishability against chosen plaintext attacks (IND-CPA) and security against selective-opening attacks (SOA). Lossy encryption can be constructed from lossy trapdoor functions [6], from smooth projective hashing [17], and also from various concrete number-theoretic and lattice-based assumptions [17].

Xue *et al.* introduced the notion of lossy trapdoor relations (LTDR) in 2014 [25], which is a relaxation of LTDF for it does not require the recovery of the pre-image, thus is generally more efficient. It was shown in [25] that LTDR is useful in constructing lossy encryption, and adaptive trapdoor relation, which is

S. Hong and J.H. Park (Eds.): ICISC 2016, LNCS 10157, pp. 126–144, 2017.
DOI: 10.1007/978-3-319-53177-9_6

a building block for chosen-ciphertext security. And in [25] LTDR is constructed from several concrete assumptions such as discrete logarithm related assumptions and subgroup membership assumptions.

Typically, lossy primitives works in two computationally indistinguishable modes: the injective mode and the lossy mode. In the injective mode an output is usually mapped from one pre-image, and this makes the primitives information-theoretically invertible. While in the lossy mode, an output corresponds to various pre-images, thus it statistically loses some information of the input.

Hybrid Encryption. Hybrid encryption, proposed by Cramer and Shoup in [11,13], is the combination of an asymmetric key encapsulation mechanism (KEM) and a symmetric data encapsulation mechanism (DEM). The KEM takes a public key and a randomness as input, outputs the first part of the ciphertext, and generates the encryption of a random encapsulated key via a key derivation function (KDF); the DEM encrypts the plaintext with the encapsulated key, and outputs the second part of the ciphertext. Given the secret key of the KEM part and the ciphertext, both the encapsulated key and the plaintext can be recovered.

A hybrid encryption scheme is essentially a public key encryption scheme. Compared with general-purpose public-key encryption, hybrid encryption enjoys the advantage of unrestricted message space, and is usually more efficient, as pointed out in [13]. Regarding to security, by a composition theorem, it is proved that a secure KEM plus a secure DEM can yield a secure hybrid encryption [13]. Thus, the KEM-DEM paradigm allows us to separate the design of the two parts. In many cases a simple one-time pad is enough for the DEM part, and we can focus on the KEM part.

However, whether deterministic public key encryption (DPKE), which is a promising solution to the issues of searchable encryption and randomness-subversion [1–3,8,9,16,21,23], can benefit from the KEM-DEM paradigm is a long-pending problem. Since in DPKE the encryption algorithm is deterministic, there is no randomness for generating the encapsulated key in the KEM. In [3] a hybrid encryption style DPKE was proposed, with an LTDF playing the KEM part, and a one-time pad playing the DEM part. Since an LTDF statistically hides the information of its pre-image in the lossy mode, it can cooperate with a powerful KDF, the universal computational extractor for statistically unpredictable sources $\mathsf{UCE}[\mathsf{S}^{\mathsf{sup}}]$, which is a strong primitive introduced by Bellare *et al.* in [4,5] and is an important tool in the DPKE construction of [3].

Motivated by the usefulness of previous lossy primitives and the advantage of hybrid encryption, it is interesting to enhance the notion of KEM with lossiness, which is a natural match of the newly proposed primitive $\mathsf{UCE}[\mathsf{S}^{\mathsf{sup}}]$, as stated in [3]. Also, it is natural to generalize the KEM usage of LTDF in [3] to embrace more efficient constructions from other primitives.

1.1 Our Contributions: Lossy KEM

Definition. We define a new lossy primitive called lossy key encapsulation mechanism, which extends the usage of several lossy primitives in some scenarios, e.g. LTDF and LTDR, to the form of KEM.

Originally, the syntax of KEM requires that the encapsulation algorithm generate a ciphertext C and an encapsulated key K out of an input randomness r. Generally, looking inside, the encapsulation algorithm can be decomposed into two subroutines: one generates a binary relation (C, tK), where C is the ciphertext, and tK is the material for producing the encapsulated key K and is usually obtained by applying an injective map on r; the other is the key derivation function, which takes tK as input and outputs K. Typically the relation (C, tK) is one-way, i.e., given a random C and the public key, it is hard to find tK. Also the decapsulation algorithm can be decomposed into two subroutines: the first one recovers tK from C with the secret key, and the other is the KDF. Note that this viewpoint on KEM was implicit in [22] by Wee, with the relation being injective, that is, there exists at most one tK corresponding to C.

The syntax of lossy KEM is similar to that of the original KEM. However, akin to previous lossy primitives, lossy KEM also works in two modes, an injective mode for functionality and a lossy mode for the security proof. In the injective mode, the key material tK can be recovered from the ciphertext C with the secret key, thus K can be recovered; while in the lossy mode, the ciphertext C statistically hides the information of tK and the encapsulated key K. The injective mode and the lossy mode should be computationally indistinguishable given the public key. We show that lossy KEM implies IND-secure KEM, just like lossy encryption implies IND-CPA secure encryption.

Constructions. Then we show the general ideas of constructing lossy KEM from two previous lossy primitives, i.e., LTDF, LTDR, and from entropic projective hashing [18]. Details of the constructions are in Sect. 4.

- Given an LTDF f, the lossy KEM on input r generates the relation $(C = f(r), tK = r)$, derives the key $K = h(tK)$, and outputs (C, K), where the KDF h is randomly chosen from a family of pairwise independent hash functions. The KDF can also be other suitable primitives. Note that the KEM usage of LTDF in the DPKE construction of [3] is just the case, with the KDF being picked from a family of $\mathsf{UCE}[\mathsf{S}^{\mathsf{sup}}]$-secure hash functions. The lossiness of the KEM follows from that of the LTDF, i.e., in the lossy mode, $C = f(r)$ statistically hides the information of r.
- Given an LTDR (f, H), where H is a publicly computable injective map, the lossy KEM on input r generates the relation $(C = f(r), tK = H(r))$, derives the key $K = h(tK)$ and outputs (C, K), where the KDF h is also randomly chosen from a family of pairwise independent hash functions.
- Given an entropic projective hashing $(H, \Lambda, R, X, L, \Pi, S, \alpha)$, where H is the private evaluation algorithm, Λ is the public evaluation algorithm, X is a language and L is a subset of X. With a public key $x \in X$, the lossy KEM on

input $r \in R$ generates the relation $(C = \alpha(r) \in S, tK = H(r, x) \in \Pi)$, derives the key $K = h(tK)$, and outputs (C, K), where α is a projective map, and h is randomly chosen from a family of pairwise independent hash functions. If $x \in L$ then the lossy KEM is working in the injective mode, otherwise if $x \in X \backslash L$ then the lossy KEM is working in the lossy mode.

Applications. The new lossy primitive lossy KEM is useful in constructing lossy encryption and deterministic public key encryption.

- With lossy KEM, we generalize constructions of lossy encryption based on LTDF and LTDR in [6] and [25] respectively, and the construction of lossy encryption from smooth projective hashing in [17].
- Moreover, we generalize the deterministic public key encryption based on LTDF in [3]. Generally, if we choose a lossy KEM constructed from LTDR, then we can get better efficiency, compared to [3].

Organization. In Sect. 2, some notations and definitions are introduced. In Sect. 3, the definition of lossy KEM is given. In Sect. 4, several constructions of lossy KEM are shown. In Sect. 5, we construct a lossy encryption from lossy KEM. In Sect. 6, we construct a DPKE from lossy KEM. Section 7 is the conclusion.

2 Preliminaries

Notations. Let λ be the security parameter. For a string x, $|x|$ denotes its length. For a finite set S, $|S|$ denotes its size. Vectors are denoted by boldface characters. For a vector $\mathbf{x}$, $|\mathbf{x}|$ denotes the number of its components. $x \xleftarrow{\$} S$ means that x is chosen from the set S uniformly at random. For a randomized algorithm A, $x \xleftarrow{\$} A(\cdot)$ means that x is assigned the output of A. An algorithm is efficient if it runs in polynomial time in its input length. A function $f(\lambda)$ is negligible if it decreases faster than any polynomial, and is denoted as $f(\lambda) \leq \epsilon(\lambda)$. The min-entropy of a random variable X is denoted as $\mathbf{H}_\infty(X) = -\log(\max_x P_X(x))$, wherein $P_X(x) = \Pr[X = x]$. Given a random variable Y, the conditional min-entropy of X is $\tilde{\mathbf{H}}_\infty(X|Y) = -\log(\mathop{\mathbf{E}}_{y \leftarrow Y} \max_x \Pr[X = x|Y = y])$ [14]. The statistical distance between two random variables X and Y is $\Delta(X, Y) = \frac{1}{2} \sum_x |P_X(x) - P_Y(x)|$, X and Y are statistically close if $\Delta(X, Y)$ is negligible, and is denoted as $X \overset{s}{\approx} Y$. X and Y are computationally indistinguishable if no efficient algorithm can tell them apart given only oracle access, and is denoted as $X \overset{c}{\approx} Y$. PPT is the short form of probabilistic polynomial time. $\perp$ is the empty symbol.

2.1 Key Encapsulation Mechanism

Here we recall the definition and security notion of KEM. In the definition we also use an alternative description, for the sake of better description of lossy KEM in subsequent sections. We believe that the alternative description is still without loss of generality and gives better understanding of KEM. We use the alternative description in some occasions if necessary. xue2014lossy A key encapsulation mechanism KEM is a triple of algorithms $(\mathsf{KEM.Kg}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec})$:

- Key generation: $(pk, sk) \xleftarrow{\$} \mathsf{KEM.Kg}(\lambda)$.
- Encapsulation: $(C, K) \leftarrow \mathsf{KEM.Enc}(pk, r)$. $\mathsf{KEM.Enc}$ can be decomposed into two subroutines, Rg and KDF.
  - Relation generation: $(C, tK) \leftarrow \mathsf{KEM.Enc.Rg}(pk, r)$, where tK is induced by an injective function of r.
  - Key derivation: $K \leftarrow \mathsf{KEM.Enc.KDF}(tK)$, where $\mathsf{KDF} : \{0,1\}^* \rightarrow \{0,1\}^*$ is a key derivation function (usually a pairwise independent hash function with its key specified in pk and sk).
- Decapsulation: $K \leftarrow \mathsf{KEM.Dec}(sk, C)$. Similarly, $\mathsf{KEM.Dec}$ can also be decomposed into two subroutines, Inv and KDF.
  - Inversion: $tK \leftarrow \mathsf{KEM.Dec.Inv}(sk, C)$;
  - Key derivation: $K \leftarrow \mathsf{KEM.Dec.KDF}(tK)$.

The IND security of KEM is described by the following game, where A is the adversary, $\mathsf{RSp}(\lambda)$ is the randomness space, $\mathsf{KEM.kl}$ is the length of the encapsulated key. $b' \stackrel{?}{=} b$ is a predicate denoting whether the two bits are equal, 1 is true and 0 is false.

$$\mathsf{Game}^{\mathsf{ind}}_{\mathsf{KEM},\mathsf{A}}(\lambda)$$
$$(pk, sk) \xleftarrow{\$} \mathsf{KEM.Kg}(\lambda); r \xleftarrow{\$} \mathsf{RSp}(\lambda); (C, K_0) \leftarrow \mathsf{KEM.Enc}(pk, r);$$
$$K_1 \xleftarrow{\$} \{0,1\}^{\mathsf{KEM.kl}}; b \xleftarrow{\$} \{0,1\}; b' \xleftarrow{\$} \mathsf{A}(pk, C, K_b); \text{Return } (b' \stackrel{?}{=} b)$$

The advantage of A in winning the game is defined as $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{KEM},\mathsf{A}}(\lambda) = 2\Pr[\mathsf{Game}^{\mathsf{ind}}_{\mathsf{KEM,A}}(\lambda)] - 1$, where $\mathsf{Game}^{\mathsf{ind}}_{\mathsf{KEM},\mathsf{A}}(\lambda)$ is the abbreviation for "$\mathsf{Game}^{\mathsf{ind}}_{\mathsf{KEM},\mathsf{A}}(\lambda) \Rightarrow 1$". The kind of abbreviation will be used throughout the paper. We say that KEM is IND secure if for all PPT adversary A, $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{KEM},\mathsf{A}}(\lambda)$ is negligible.

2.2 Lossy Primitives

Here is a brief recap of the definitions of previous lossy primitives in literatures [6,20,25].

Lossy Trapdoor Functions. A collection of (m, l)-lossy trapdoor functions is a 4-tuple of PPT algorithms $\mathsf{F} = (\mathsf{F.Ig}, \mathsf{F.Lg}, \mathsf{F.Ev}, \mathsf{F.Inv})$ described below.

- Sampling the injective mode: $(\sigma_I, \tau) \xleftarrow{\$} \mathsf{F.Ig}(\lambda)$, where σ_I is a function index, and τ is a trapdoor.
- Sampling the lossy mode: $(\sigma_L, \perp) \xleftarrow{\$} \mathsf{F.Lg}(\lambda)$. In the lossy mode, the function is irreversible, thus there is no trapdoor.
- Evaluation: $y \leftarrow \mathsf{F.Ev}(\sigma, x)$, where σ is a function index, $x \in \{0,1\}^m$. There are:
  - injective mode: if σ is produced by $\mathsf{F.Ig}(\cdot)$, then the function $\mathsf{F.Ev}(\cdot)$ is injective.
  - lossy mode: if σ is produced by $\mathsf{F.Lg}(\cdot)$, then the size of the image of $\mathsf{F.Ev}(\cdot)$ is at most 2^{m-l}, i.e., there are many pre-images corresponding to an image.
- Inversion: $x \leftarrow \mathsf{F.Inv}(\tau, y)$, i.e., the function can be inverted in the injective mode with the trapdoor.

The function indices σ_I and σ_L respectively produced in the injective mode and the lossy mode should be computationally indistinguishable.

Lossy Trapdoor Relations. A collection of (m, l)-lossy trapdoor relations is a 4-tuple of PPT algorithms $\mathsf{F} = (\mathsf{F.Ig}, \mathsf{F.Lg}, \mathsf{F.Ev}, \mathsf{F.Inv})$ described below.

- Sampling the injective mode: $(\sigma_I, \tau, H) \xleftarrow{\$} \mathsf{F.Ig}(\lambda)$, where σ_I is a function index, τ is a trapdoor, and H is a publicly computable injective map.
- Sampling the lossy mode: $(\sigma_L, \perp, H) \xleftarrow{\$} \mathsf{F.Lg}(\lambda)$. Also, there is no trapdoor in the lossy mode of LTDR.
- Encapsulation: $(y, z) \leftarrow \mathsf{F.Ev}(\sigma, H, x)$, where $x \in \{0,1\}^m$, $y = f(\sigma, x)$ for a function f parameterized by σ, $z = H(x)$, and there are:
  - injective mode: if σ is produced by $\mathsf{F.Ig}(\cdot)$, then the function $f(\sigma, \cdot)$ is injective.
  - lossy mode: if σ is produced by $\mathsf{F.Lg}(\cdot)$, then the size of the image of $f(\sigma, \cdot)$ is at most 2^{m-l}.
- Decapsulation: $z \leftarrow \mathsf{F.Inv}(\tau, H, y)$, where $z = H(x)$. That is, the relation (y, z) can be recovered in the injective mode given the trapdoor.

Also, the function indices σ_I and σ_L respectively produced in the injective mode and the lossy mode should be computationally indistinguishable. LTDR is generally more efficient than LTDF since it does not require the recovery of the pre-image x but a publicly computable injective map of it, i.e., $z = H(x)$, as shown in [25].

Lossy Encryption. A lossy public key encryption LE is a 4-tuple of algorithms, $(\mathsf{LE.Kg}, \mathsf{LE.LKg}, \mathsf{LE.Enc}, \mathsf{LE.Dec})$.

- Key generation: $(pk_I, sk) \xleftarrow{\$} \mathsf{LE.Kg}(\lambda)$.

- Lossy key generation: $(pk_L, \perp) \xleftarrow{\$} \mathsf{LE.LKg}(\lambda)$.
- Encryption: $C \leftarrow \mathsf{LE.Enc}(pk, m, r)$, where m is the plaintext, and r is the randomness.
- Decryption: $m \leftarrow \mathsf{LE.Dec}(sk, C)$.

And the algorithms should satisfy the following addition properties:

1. Correctness: for all (pk_I, sk) generated by $\mathsf{LE.Kg}$, all m and r, there is $m = \mathsf{LE.Dec}(sk, C)$ where $C \leftarrow \mathsf{LE.Enc}(pk_I, m, r)$.
2. Lossiness: for all pk_L generated by $\mathsf{LE.LKg}$, and any pair of distinct messages (m_0, m_1), the respective distributions of the ciphertexts of m_0 and m_1 are statistically close, i.e., $\mathsf{LE.Enc}(pk_L, m_0, R) \overset{s}{\approx} \mathsf{LE.ENC}(pk_L, m_1, R)$, where R is the randomness space.
3. Indistinguishability: The public keys pk_I and pk_L respectively generated by $\mathsf{LE.Kg}$ and $\mathsf{LE.LKg}$ are computationally indistinguishable.

2.3 Entropic Projective Hashing

Cramer and Shoup introduced smooth projective hashing (SPH) in [12]. SPH is a family of keyed hash functions defined over a "hard" language, and is useful in building many cryptographic primitives such as chosen-ciphertext secure encryption, leakage-resilient encryption, lossy encryption. In [18] Kiltz *et al.* generalized the smoothness property of SPH to "κ-entropic". A κ-entropic projective hashing $\mathsf{P} = (H, \Lambda, R, X, L, \Pi, S, \alpha)$ is explained below:

- Hard language (X, L): X is a language and L is a subset of X. For any $x \in L$ there is a witness w, and for $x \in X \backslash L$ there is no witness. By assumption, it is hard to distinguish $x \in L$ and $x' \in X \backslash L$ efficiently.
- Key Projection α: The hash function is keyed by $r \in R$. There is also a projective map $\alpha : R \mapsto S$, given a hash key $r \in R$, generates a projective key $s = \alpha(r) \in S$. Both r and s can be used to evaluate the hash value, in the private evaluation algorithm H and public evaluation algorithm Λ respectively.
- Private evaluation H: Given the hash key r, and a hash input $x \in X$, the hash value $\pi = H(r, x) \in \Pi$ is efficiently computable.
- Public evaluation Λ: The public evaluation algorithm Λ only works for $x \in L$. For any hash key $r \in R$, the action of $H(r, \cdot)$ on L is completed determined by $\alpha(r)$. That is, for any $x \in L$ with witness w, Λ correctly computes the hash value with w and $\alpha(r)$, i.e., $\Lambda(\alpha(r), x, w) = H(r, x)$. It is also called the projective property.
- κ-entropic property: The property is defined for $x \in X \backslash L$. P is ϵ-almost κ-entropic if for all $x \in X \backslash L$, there is $\Pr[\tilde{\mathbf{H}}_\infty(H(r, x)) | \alpha(r) \geq \kappa] \geq 1 - \epsilon$. That is, the hash value of an input $x \in X \backslash L$ cannot be determined given the projective key in the information-theoretic sense.

P is smooth if the two distributions over $X \backslash L \times S \times \Pi$, defined as $Z_1 = (x, s = \alpha(r), \pi = H(r, x))$ and $Z_2 = (x, s = \alpha(r), \pi')$ where $r \in R$ and $\pi' \xleftarrow{\$} \Pi$,

are statistically close [12]. That is, for $x \in X \backslash L$, the hash value $H(r, x)$ is nearly uniformly distributed in its range Π given only the projective key $\alpha(r)$. Obviously smoothness is stronger than the κ-entropic property. However, κ-entropic is enough in many scenarios. And as shown in [18], the κ-entropic property can be converted into smoothness with a pairwise independent hash.

3 Lossy Key Encapsulation Mechanism

In this section we define the notion of lossy key encapsulation mechanism. The definition combines those of KEM and lossy encryption.

Definition 1 (Lossy Key Encapsulation Mechanism). *A lossy key encapsulation mechanism* LKE *is a 4-tuple of algorithms,* $(\mathsf{LKE.Kg}, \mathsf{LKE.LKg}, \mathsf{LKE.Enc}, \mathsf{LKE.Dec})$.

- *Key generation:* $(pk_I, sk) \xleftarrow{\$} \mathsf{LKE.Kg}(\lambda)$.
- *Lossy key generation:* $(pk_L, \perp) \xleftarrow{\$} \mathsf{LKE.LKg}(\lambda)$.
- *Encapsulation:* $(C, K) \leftarrow \mathsf{LKE.Enc}(pk, r)$*, where* pk *is generated by either* $\mathsf{LKE.Kg}$ *or* $\mathsf{LKE.LKg}$. $\mathsf{LKE.Enc}$ *can be decomposed into two subroutines,* LRg *and* KDF.
  - *Lossy relation generation:* $(C, tK) \leftarrow \mathsf{LKE.Enc.LRg}(pk, r)$*, where* C *is the image of* r*, and* tK *is induced by an injective function from* r.
  - *Key derivation:* $K \leftarrow \mathsf{LKE.Enc.KDF}(tK)$*, where* $\mathsf{KDF} : \{0,1\}^* \rightarrow \{0,1\}^*$ *is a key derivation function with its key specified in* pk *and* sk.
- *Decapsulation:* $K \leftarrow \mathsf{LKE.Dec}(sk, C)$*. Similarly,* $\mathsf{LKE.Dec}$ *can also be decomposed into two subroutines,* Inv *and* KDF.
  - *Inversion:* $tK \leftarrow \mathsf{LKE.Dec.Inv}(sk, C)$*;*
  - *Key derivation:* $K \leftarrow \mathsf{LKE.Dec.KDF}(tK)$*;*

We require the following properties for the algorithms:

1. *Correctness: for all* (pk, sk) *generated by* $\mathsf{LKE.Kg}$*, there is* $K = \mathsf{LKE.Dec}(sk, C)$ *where* $(C, K) \leftarrow \mathsf{LKE.Enc}(pk, r)$.
2. *Lossiness: for all* pk *generated by* $\mathsf{LKE.LKg}$*,* $(C, K) \leftarrow \mathsf{LKE.Enc}(pk, r)$*,* C *statistically hides the information of* r *and consequently the information of* tK*, thus* K *can not be recovered. In detail, denote the size of the set of all* $r's$ *as* $2^{\mathsf{LKE.il}}$ *and the size of the set of all* $C's$ *as* $2^{\mathsf{LKE.cl}}$*, then* $\mathsf{LKE.cl} < \mathsf{LKE.il}$*. We call* $\delta = \mathsf{LKE.il} - \mathsf{LKE.cl}$ *the lossiness of* LKE*, and there is* $\tilde{\mathbf{H}}_\infty(tK|C) \geq \delta$.
3. *Indistinguishability: No polynomial time algorithm can distinguish the public keys generated by* $\mathsf{LKE.Kg}$ *and* $\mathsf{LKE.LKg}$*. We further describe the requirement by the following game:*

$$\mathsf{Game}^{\mathsf{loss}}_{\mathsf{LKE},\mathsf{A}}(\lambda)$$
$$(pk_0, sk) \xleftarrow{\$} \mathsf{LKE.Kg}(\lambda); (pk_1, \perp) \xleftarrow{\$} \mathsf{LKE.LKg}(\lambda);$$
$$b \xleftarrow{\$} \{0,1\}; b' \xleftarrow{\$} \mathsf{A}(\lambda, pk_b);\ \textit{Return}\ (b' \stackrel{?}{=} b)$$

The advantage of the adversary A *in winning the game, defined as* $\mathsf{Adv}^{\mathsf{loss}}_{\mathsf{LKE},\mathsf{A}}(\lambda) = 2\Pr[\mathsf{Game}^{\mathsf{loss}}_{\mathsf{LKE},\mathsf{A}}(\lambda)] - 1$*, is negligible.*

Akin to the case that lossy encryption implies IND-CPA secure encryption [6], lossy KEM also implies IND-secure KEM. With a generalized leftover hash lemma proposed by Dodis *et al.* in [19], we prove that a lossy KEM is IND secure with the key derivation function KDF being chosen from a family of pairwise independent hash functions.

Lemma 1 (Generalized Leftover Hash Lemma [19]). *Let X, Y be random variables such that $X \in D$ and $\tilde{\mathbf{H}}_\infty(X|Y) \geq \delta$. Let $\mathcal{H}$ be a family of pairwise independent hash function from D to $\{0,1\}^k$. Then for $h \xleftarrow{\$} \mathcal{H}$, and $k \leq \delta - 2\log(1/\epsilon)$ there is $\Delta((Y, h, h(X)), (Y, h, U_k)) \leq \epsilon$.*

Theorem 1. *Assume that the key derivation function KDF is randomly chosen from a family of pairwise independent hash functions mapping D to $\{0,1\}^k$, where D is the set of all tK's and $k \leq \delta - 2\log(1/\epsilon)$, then a lossy KEM LKE with lossiness δ is also IND secure. Specifically, let A be an IND adversary, then we could construct a lossy KEM adversary B, such that for A, B, there is $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LKE},\mathsf{A}}(\cdot) \leq 2\mathsf{Adv}^{\mathsf{loss}}_{\mathsf{LKE},\mathsf{B}}(\cdot)$.*

Proof. We prove the theorem via a sequence of games. Let A be an IND adversary attacking the IND security of the lossy KEM, and Game_0 be the original IND game. Denote the probability of A in winning Game_i as $\Pr[\mathsf{G}^{\mathsf{A}}_i(\cdot)]$, then $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LKE},\mathsf{A}}(\cdot) = 2\Pr[\mathsf{G}^{\mathsf{A}}_0(\cdot) - 1]$.

Game_1: Replace the key generation algorithm $\mathsf{LKE.Kg}(\cdot)$ with $\mathsf{LKE.LKg}(\cdot)$. Then we can construct a lossy KEM adversary B invoking A as follows:

$\mathsf{B}(\lambda, pk_b)$

$r \xleftarrow{\$} \mathsf{RSp}(\lambda); (C, tK) \leftarrow \mathsf{LKE.Enc.LRg}(pk, r);$

$K_0 \leftarrow \mathsf{LKE.Enc.KDF}(tK); K_1 \xleftarrow{\$} \{0,1\}^k; d \xleftarrow{\$} \{0,1\};$

$d' \xleftarrow{\$} \mathsf{A}(pk_b, \mathbf{C}, K_d);$ If $(d' = d)$ return 0, otherwise return 1.

If $b = 0$, i.e., B receives a normal public key, then B is simulating Game_0 for A. Else, if $b = 1$, i.e., B receives a lossy public key, then B is simulating Game_1. Hence, $\Pr[\mathsf{G}^{\mathsf{A}}_0(\cdot)] - \Pr[\mathsf{G}^{\mathsf{A}}_1(\cdot)] \leq \mathsf{Adv}^{\mathsf{loss}}_{\mathsf{LKE},\mathsf{B}}(\cdot)$.

In Game_1, LKE is working in the lossy mode, thus the ciphertext C statistically hides the information of tK, i.e., $\tilde{\mathbf{H}}_\infty(tK|C) \geq \delta$. With Lemma 1 there is $\Delta((C, \mathsf{KDF}, \mathsf{KDF}(tK)), (C, \mathsf{KDF}, U_k)) \leq \epsilon$, i.e., K_0 and K_1 are statistically close, thus the probability of A in winning the game is $\Pr[\mathsf{G}^1_{\mathsf{A}}(\cdot)] = 1/2$.

By summing up there is $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LKE},\mathsf{A}}(\cdot) \leq 2\mathsf{Adv}^{\mathsf{loss}}_{\mathsf{LKE},\mathsf{B}}(\cdot)$, which is negligible since LKE is assumed to be lossy. □

4 Constructions of Lossy KEM

Here we show constructions of lossy KEM from lossy trapdoor functions, lossy trapdoor relations, and entropic projective hashing. The constructions are direct and simple.

4.1 Lossy KEM from LTDF

Given a collection of lossy trapdoor functions $\mathsf{F} = (\mathsf{F.Ig}, \mathsf{F.Lg}, \mathsf{F.Ev}, \mathsf{F.Inv})$, and a family of pairwise independent hash functions $\mathcal{H}$, we could construct a lossy KEM $\mathsf{LK} = (\mathsf{LK.Kg}, \mathsf{LK.LKg}, \mathsf{LK.Enc}, \mathsf{LK.Dec})$ as follows:

- Key generation $(pk_I, sk) \xleftarrow{\$} \mathsf{LK.Kg}(\lambda)$: $(\sigma_I, \tau) \xleftarrow{\$} \mathsf{F.Ig}(\lambda)$; $h \xleftarrow{\$} \mathcal{H}$; $(pk_I, sk) \leftarrow ((\sigma_I, h), (\tau, h))$.
- Lossy key generation $(pk_L, \perp) \xleftarrow{\$} \mathsf{LK.LKg}(\lambda)$: $(\sigma_L, \perp) \xleftarrow{\$} \mathsf{F.Lg}(\lambda)$; $h \xleftarrow{\$} \mathcal{H}$; $(pk_L, \perp) \leftarrow ((\sigma_L, h), \perp)$.
- Encapsulation $(C, K) \leftarrow \mathsf{LK.Enc}(pk, r)$: $C \leftarrow \mathsf{F.Ev}(\sigma, r), tK \leftarrow r, K \leftarrow h(tK)$.
- Decapsulation $K \leftarrow \mathsf{LK.Dec}(sk, C)$: $r \leftarrow \mathsf{F.Inv}(\tau, C), tK \leftarrow r, K \leftarrow h(tK)$.

Theorem 2. *Assume that* F *is a collection of* (m, l)*-lossy trapdoor functions, and* $\mathcal{H}$ *is a family of pairwise independent hash functions, then* LK *is a lossy KEM with lossiness* $\delta = l$*.*

Proof.
- Correctness: follows from the injective mode of F, i.e., for all (σ_I, τ) produced by $\mathsf{F.Ig}$, and $C \leftarrow \mathsf{F.Ev}(\sigma_I, r)$, there is $r = \mathsf{F.Inv}(\tau, C)$, thus $K \leftarrow h(r)$ can be recovered.
- Lossiness: follows from the lossy mode of F, i.e., for all $(\sigma_L, \perp)$ produced by $\mathsf{F.Lg}$, and $C \leftarrow \mathsf{F.Ev}(\sigma_L, r)$, the size of the set of all C's is at most 2^{m-l}, i.e., C statistically loses at least l bits information of r. Since $tK = r$, it means that $\tilde{\mathbf{H}}_\infty(tK|C) \geq l$. Thus $K = h(r)$ can not be recovered. And the lossiness of LK is l.
- Indistinguishability: follows from the indistinguishability of the injective mode and the lossy mode of F.

□

4.2 Lossy KEM from LTDR

Given a collection of lossy trapdoor relations $\mathsf{F} = (\mathsf{F.Ig}, \mathsf{F.Lg}, \mathsf{F.Ev}, \mathsf{F.Inv})$ and and a family of pairwise independent hash functions $\mathcal{H}$, we could construct a lossy KEM $\mathsf{LK} = (\mathsf{LK.Kg}, \mathsf{LK.LKg}, \mathsf{LK.Enc}, \mathsf{LK.Dec})$ as follows:

- Key generation $(pk_I, sk) \xleftarrow{\$} \mathsf{LK.Kg}(\lambda)$: $(\sigma_I, H, \tau) \xleftarrow{\$} \mathsf{F.Ig}(\lambda)$; $h \xleftarrow{\$} \mathcal{H}$; $(pk_I, sk) \leftarrow ((\sigma_I, H, h), (\tau, H, h))$.
- Lossy key generation $(pk_L, \perp) \xleftarrow{\$} \mathsf{LK.LKg}(\lambda)$: $(\sigma_L, H, \perp) \xleftarrow{\$} \mathsf{F.Lg}(\lambda)$; $h \xleftarrow{\$} \mathcal{H}$; $(pk_L, \perp) \leftarrow ((\sigma_L, H, h), \perp)$.
- Encapsulation $(C, K) \leftarrow \mathsf{LK.Enc}(pk, r)$: $(C, H(r)) \leftarrow \mathsf{F.Ev}(\sigma, H, r), tK \leftarrow H(r), K \leftarrow h(tK)$.
- Decapsulation $K \leftarrow \mathsf{LK.Dec}(sk, C)$: $H(r) \leftarrow \mathsf{F.Inv}(\tau, H, C), tK \leftarrow H(r), K \leftarrow h(H(r))$.

Theorem 3. *Assume that* F *is a collection of* (m, l)*-lossy trapdoor relations, and* $\mathcal{H}$ *is a family of pairwise independent hash functions, then* LK *is a lossy KEM with lossiness* $\delta = l$*.*

Proof. – Correctness: follows from the injective mode of F, i.e., for all (σ_I, τ) produced by $\mathsf{F.Ig}$, and $(C, H(r)) \leftarrow \mathsf{F.Ev}(\sigma_I, H, r)$, there is $H(r) = \mathsf{F.Inv}(\tau, H, C)$, thus $K \leftarrow h(H(r))$ can be recovered.
– Lossiness: follows from the lossy mode of F, i.e., for all $(\sigma_L, \perp)$ produced by $\mathsf{F.Lg}$, and $C \leftarrow \mathsf{F.Ev}(\sigma_L, H, r)$, the size of the set of all C's is at most 2^{m-l}, i.e., C statistically loses at least l bits information of r and $H(r)$. Since $tK = H(r)$, there is $\tilde{\mathbf{H}}_\infty(tK|C) \geq l$. Thus $K = h(H(r))$ can not be recovered. And the lossiness of LK is l.
– Indistinguishability: follows from the indistinguishability of the injective mode and the lossy mode of F.

□

4.3 Lossy KEM from Entropic Projective Hashing

In [23] Wee defined dual projective hashing, which is similar to smooth projective hashing, for the purpose of constructing lossy trapdoor function and deterministic public key encryption. Here we show that lossy KEM can be directly constructed from the weaker primitive, entropic projective hashing, without making a detour from lossy trapdoor functions, in a similar way with the lossy encryption constructed from smooth projective hashing in [17].

Given a κ-entropic projective hashing $\mathsf{P} = (H, \Lambda, R, X, L, \Pi, S, \alpha)$ and a family of pairwise independent hash functions $\mathcal{H}$, we construct a lossy KEM $\mathsf{LK} = (\mathsf{LK.Kg}, \mathsf{LK.LKg}, \mathsf{LK.Enc}, \mathsf{LK.Dec})$ as follows:

– Key generation $(pk_I, sk) \xleftarrow{\$} \mathsf{LK.Kg}(\lambda)$: $(x, w) \xleftarrow{\$} L$; $h \xleftarrow{\$} \mathcal{H}$; $(pk_I, sk) \leftarrow ((x, h), (x, w, h))$.
– Lossy key generation $(pk_L, \perp) \xleftarrow{\$} \mathsf{LK.LKg}(\lambda)$: $(x', \perp) \xleftarrow{\$} X \backslash L$; $h \xleftarrow{\$} \mathcal{H}$; $(pk_L, \perp) \leftarrow ((x', h), \perp)$.
– Encapsulation $(C, K) \leftarrow \mathsf{LK.Enc}(pk, r)$: $C \leftarrow \alpha(r), tK \leftarrow H(r, x), K \leftarrow h(tK)$.
– Decapsulation $K \leftarrow \mathsf{LK.Dec}(sk, C)$: $tK \leftarrow \Lambda(\alpha(r), x, w), K \leftarrow h(tK)$.

Theorem 4. *Assume that* P *is a* κ*-entropic projective hashing, and* $\mathcal{H}$ *is a family of pairwise independent hash functions, then* LK *is a lossy KEM with lossiness* κ*.*

Proof. – Correctness: Follows from the projective property of P, i.e., for all $x \in L$ with witness w, and $C = \alpha(r)$, there is $tK = \Lambda(\alpha(r), x, w) = H(r, x)$, thus $K = h(tK)$ can be recovered.
– Lossiness: Follows from the entropic property of P, since for all $x' \in X \backslash L$, given $C = \alpha(r)$, $tK = H(r, x')$ can not be determined by C, and with overwhelming probability there is $\tilde{\mathbf{H}}_\infty(H(r, x')|\alpha(r)) \geq \kappa$. It means that $H(\cdot, x')$ is an injective function of r in the case of $x' \in X \backslash L$, and C statistically hides the information of tK, with lossiness κ.
– Indistinguishability: Follows from the indistinguishability of $x \in L$ and $x' \in X \backslash L$.

□

5 Lossy Encryption from Lossy KEM

A natural and immediate application of lossy KEM is to construct lossy encryption, with a proper randomness extractor, e.g., a pairwise-independent hash, being the key derivation function. In detail, given a lossy KEM $\mathsf{LKE} = (\mathsf{LKE.Kg}, \mathsf{LKE.LKg}, \mathsf{LKE.Enc}, \mathsf{LKE.Dec})$, with its encapsulated key length being k; let the KDF h of LKE be chosen from a family of pairwise-independent hash functions $\mathcal{H}$ with proper i/o length, and the description of h be specified in the public key and secret key. Then we construct a lossy encryption scheme $\mathsf{LE} = (\mathsf{LE.Kg}, \mathsf{LE.LKg}, \mathsf{LE.Enc}, \mathsf{LE.Dec})$ encrypting messages from $\{0,1\}^k$as follows:

$\mathsf{LE.Kg}(1^\lambda)$
$(pk_I, sk) \xleftarrow{\$} \mathsf{LKE.Kg}(1^\lambda)$
$(PK_I, SK) \leftarrow (pk_I, sk)$
Return (PK_I, SK)

$\mathsf{LE.LKg}(1^\lambda)$
$(pk_L, \perp) \xleftarrow{\$} \mathsf{LKE.LKg}(1^\lambda)$
$PK_L \leftarrow pk_L$
Return $(PK_L, \perp)$

$\mathsf{LE.Enc}(PK, m, r)$
$(C_1, tK) \leftarrow \mathsf{LKE.Enc.Rg}(pk, r)$
$K \leftarrow h(tK)$
$C_2 \leftarrow m \oplus K$
Return (C_1, C_2)

$\mathsf{LE.Dec}(SK, C)$
$(C_1, C_2) \leftarrow C$
$tK \leftarrow \mathsf{LKE.Dec.Inv}(SK, C_1)$
$K \leftarrow h(tK)$
$m \leftarrow C_2 \oplus K$
Return m

The construction is a generalization of the lossy encryptions from lossy trapdoor functions and lossy trapdoor relations proposed in [6,25]; if LKE is constructed from entropic projective hashing, then it also generalizes the lossy encryption from smooth projective hashing in [17]; thus it is obvious that LE satisfies the properties of lossy encryption.

Theorem 5. *Assume that* LKE *is a lossy KEM with lossiness* δ*, and* $\mathcal{H}$ *is a family of pairwise independent hash functions mapping* D *to* $\{0,1\}^k$*, where* D *is the set of all* tK*'s and* $k \leq \delta - 2\log(1/\epsilon)$*. Then* LE *is a lossy encryption.*

Proof.
- The correctness and indistinguishability of LE follow readily from those properties of LKE.
- As to the lossiness, i.e., for all PK_L generated by $\mathsf{LE.LKg}$, the encryption of any pair of distinct messages (m_0, m_1) should be statistically close, it mainly follows from the lossy mode of the lossy KEM. In the lossy mode, there is $\tilde{\mathbf{H}}_\infty(tK|C_1) \geq \delta$. With Lemma 1 we know that K is statistically close to the uniform distribution on $\{0,1\}^k$. Consequently, C_2 statistically hides the information of the plaintext. Thus, the ciphertext distributions of two distinct messages are statistically close.

□

6 Deterministic Public Key Encryption from Lossy KEM

Another application of lossy KEM is the construction of deterministic public key encryption scheme. Firstly we recall some definitions.

6.1 Deterministic Public Key Encryption

A deterministic PKE scheme $\mathsf{DE} = (\mathsf{DE.Kg}, \mathsf{DE.Enc}, \mathsf{DE.Dec})$ is defined below:

1. (probabilistic) Key generation: $(PK, SK) \xleftarrow{\$} \mathsf{DE.Kg}(\lambda)$;
2. (deterministic) Encryption: $C \leftarrow \mathsf{DE.Enc}(PK, M)$;
3. (deterministic) Decryption: $M \leftarrow \mathsf{DE.Dec}(SK, C)$.

We use the IND-style definition of PRIV security from [2]. A PRIV adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ of the DPKE scheme is a pair of PPT algorithms:

- Message generator A_1: $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathsf{A}_1(\lambda)$; it is required that
  i. $|\mathbf{m}_0| = |\mathbf{m}_1| \leq v(\lambda)$ for a certain polynomial v, and $|\mathbf{m}_0[i]| = |\mathbf{m}_1[i]|$ for every $1 \leq i \leq |\mathbf{m}_0|$, and
  ii. For $i \neq j$, $1 \leq i, j \leq |\mathbf{m}_0|$, there is $\mathbf{m}_b[i] \neq \mathbf{m}_b[j]$ for $b = 0$ and $b = 1$ respectively.
- Guesser A_2: $b' \leftarrow \mathrm{A}_2(\lambda, PK, \mathbf{c}_b)$.

To make the security of DPKE schemes achievable, we should further stipulate that the adversary A have *high min-entropy*. That is, the function $\mathsf{Guess}_{\mathsf{A}}(\lambda) = \Pr[\mathbf{m}_b[i] = m : (\mathbf{m}_0, \mathbf{m}_1) \xleftarrow{\$} \mathsf{A}_1(\lambda)]$ is negligible for all $b \in \{0,1\}, 1 \leq i \leq |\mathbf{m}_b|, m \in \{0,1\}^*$.

The IND-style PRIV security is described by the following game:

$$
\begin{array}{l}
\mathsf{Game}^{\mathsf{priv}}_{\mathsf{DE},\mathsf{A}}(\lambda) \\
(pk, sk) \xleftarrow{\$} \mathsf{DE.Kg}(\lambda); b \xleftarrow{\$} \{0,1\}; (\mathbf{m}_0, \mathbf{m}_1) \xleftarrow{\$} \mathsf{A}_1(\lambda); \\
\text{For } i = 1 \text{ to } |\mathbf{m}_0| \text{ do } \mathbf{c}[i] \leftarrow \mathsf{DE.Enc}(pk, \mathbf{m}_b[i]); \\
b' \xleftarrow{\$} \mathsf{A}_2(\lambda, pk, \mathbf{c}); \text{ Return } (b' \stackrel{?}{=} b)
\end{array}
$$

The advantage of the adversary A in winning the game is defined as $\mathsf{Adv}^{\mathsf{priv}}_{\mathsf{DE},\mathsf{A}}(\lambda) = 2\Pr[\mathsf{Game}^{\mathsf{priv}}_{\mathsf{DE},\mathsf{A}}(\lambda)] - 1$.

We say that DE is PRIV secure if $\mathsf{Adv}^{\mathsf{priv}}_{\mathsf{DE},\mathsf{A}}(\cdot)$ is negligible for all PPT adversary A with high min-entropy.

6.2 Universal Computational Extractor

In [3] Bellare and Hoang solved the long-pending open problem of constructing full PRIV secure DPKE in the standard model with the "UCE + LTDF" method, where UCE stands for universal computational extractor studied in [4,5,10].

A family of hash functions H=(H.Kg,H.Ev)is UCE[S] secure if it is indistinguishable with a random oracle of the same input and output length for any PPT adversary pair (S, D), where S is called the source and D is called the distinguisher. S interacts with an oracle HASH and outputs a leakage L describing the interaction. The oracle HASH is decided by a bit $b \in \{0,1\}$. If $b = 0$ then HASH is a random oracle [7]; otherwise, HASH is a function from H. The distinguisher D receives the leakage L and outputs a guess bit about HASH. Here is the formal definition of the UCE security and the oracle HASH.

$\mathsf{Game}^{\mathsf{uce}}_{\mathsf{H},S,D}(1^\lambda)$
$b \xleftarrow{\$} \{0,1\}; hk \xleftarrow{\$} \mathsf{H.Kg}(1^\lambda);$
$L \xleftarrow{\$} S^{\mathsf{HASH}}(1^\lambda); b' \xleftarrow{\$} D(1^\lambda, hk, L);$
Return $(b' \stackrel{?}{=} b)$

$\mathsf{HASH}(x, 1^l)$
If $T[x,l] = \perp$ then
If $b = 0$ then $T[x,l] \xleftarrow{\$} \{0,1\}^l$
Else $T[x,l] \leftarrow \mathsf{H.Ev}(1^\lambda, hk, x, 1^l)$
Return $T[x,l]$

However, to make UCE security meaningful, the source S should be restricted to a certain type. In this paper we use statistically unpredictable sources, i.e., the HASH queries of S is hard to guess for a statistical predictor P given the leakage of S. Since the unpredictability of S is the property of S and is unrelated to the property of H, here the oracle HASH is the random oracle.

$\mathsf{Game}^{\mathsf{pred}}_{S,P}(1^\lambda)$
$Q \leftarrow \emptyset; L \xleftarrow{\$} S^{\mathsf{HASH}}(1^\lambda);$
$Q' \xleftarrow{\$} P(1^\lambda, L);$ Return $(Q' \cap Q \neq \emptyset)$

$\mathsf{HASH}(x, 1^l)$
If $T[x,l] = \perp$ then $T[x,l] \xleftarrow{\$} \{0,1\}^l;$
$Q \leftarrow Q \cup x;$ Return $T[x,l]$

We say that a hash family H is $\mathsf{UCE}[\mathsf{S}^{\mathsf{sup}}]$ secure if $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},\mathsf{S},\mathsf{D}} = 2\Pr[\mathsf{Game}^{\mathsf{uce}}_{\mathsf{H},\mathsf{S},\mathsf{D}}(1^\lambda)] - 1$ is negligible for all PPT adversaries (S, D), where S is statistically unpredictable for all computationally unbounded predictor P, with $\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{S},\mathsf{P}}(1^\lambda) = \Pr[\mathsf{Game}^{\mathsf{pred}}_{\mathsf{S},\mathsf{P}}(1^\lambda)]$ being negligible.

6.3 DPKE from Lossy KEM

We generalize the "UCE + LTDF" method for constructing full PRIV-secure DPKE in the standard model proposed in [3] to a "UCE + lossy KEM" way. Given a lossy KEM LKE = (LKE.Kg, LKE.LKg, LKE.Enc, LKE.Dec), with its input length denoted as LKE.il, ciphertext length denoted as LKE.cl, and encapsulated key length denoted as LKE.kl; and a $\mathsf{UCE}[\mathsf{S}^{\mathsf{sup}}]$ secure hash function family H=(H.Kg, H.Ev) with variable input/output length, we construct a deterministic public key encryption DE = (DE.Kg, DE.LKg, DE.Enc, DE.Dec) as follows:

$\mathsf{DE.Kg}(\lambda)$
$(pk, sk) \xleftarrow{\$} \mathsf{LKE.Kg}(\lambda)$
$hk \xleftarrow{\$} \mathsf{H.Kg}(\lambda)$
$PK \leftarrow (pk, hk)$
$SK \leftarrow (sk, hk)$
Return (PK, SK)

$\mathsf{DE.Enc}(PK, m)$
$r \leftarrow \mathsf{H.Ev}(hk, m, 1^{\mathsf{LKE.il}})$
$(C_1, tK) \leftarrow \mathsf{LKE.Enc.Rg}(pk, r)$
$K \leftarrow \mathsf{H.Ev}(hk, tK, 1^{\mathsf{LKE.kl}})$
$C_2 \leftarrow m \oplus K$
Return (C_1, C_2)

$\mathsf{DE.Dec}(\lambda)$
$(C_1, C_2) \leftarrow C$
$tK \leftarrow \mathsf{LKE.Dec.Inv}(sk, C_1)$
$K \leftarrow \mathsf{H.Ev}(hk, tK, 1^{\mathsf{LKE.kl}})$
$m \leftarrow C_2 \oplus K$
Return m

Then we prove the PRIV security of DE with the following theorem, which is similar to the Theorem 3.2 of [3], since the construction is a generalization of the DE1 scheme in [3].

Theorem 6. *Assume that* LKE *is a lossy KEM,* H *is a* $\mathsf{UCE}[\mathsf{S}^{\mathsf{sup}}]$ *secure hash family with variable output length, then the deterministic public key encryption* DE *is PRIV secure. Specifically, let* $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ *be a PRIV adversary with high min-entropy, then we could construct a lossy KEM adversary* B*, a pair of UCE adversary* (S, D)*, such that for* A, B *and an arbitrary statistical predictor* P*,*

$$\mathsf{Adv}^{\mathsf{priv}}_{\mathsf{DE},\mathsf{A}}(\cdot) \le 2\mathsf{Adv}^{\mathsf{loss}}_{\mathsf{LKE},\mathsf{B}}(\cdot) + 2\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},\mathsf{S},\mathsf{D}}(\cdot) + 3v^2/2^{\mathsf{LKE.il}},$$
$$\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{S},\mathsf{P}}(\cdot) \le qv\mathsf{Guess}_{\mathsf{A}}(\cdot) + 3v^2/2^{1+\mathsf{LKE.il}} + qv/2^{\delta},$$

where v *bounds the size of message vectors output by* A*,* δ *is the lossiness of* LKE*, and* q *bounds the output size of* P*.*

Proof. Let Game_0 be the original PRIV game. We prove the theorem via a sequence of games. Denote the probability of A in winning Game_i as $\Pr[\mathsf{G}^{\mathsf{A}}_i(\cdot)]$. Thus the advantage of A is $\mathsf{Adv}^{\mathsf{priv}}_{\mathsf{DE},\mathsf{A}}(\cdot) = 2\Pr[\mathsf{G}^{\mathsf{A}}_0(\cdot)] - 1$.

Game_1: Replace $\mathsf{LKE.Kg}(\cdot)$ with $\mathsf{LKE.LKg}(\cdot)$. We can construct a lossy KEM adversary B simulating a PRIV game for the adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ as follows:

$\mathsf{B}(\lambda, pk)$
$(\mathbf{m}_0, \mathbf{m}_1) \xleftarrow{\$} \mathsf{A}_1(\lambda); hk \xleftarrow{\$} \mathsf{H.Kg}(\lambda); PK \leftarrow (pk, hk); b \xleftarrow{\$} \{0,1\};$
For $i = 1$ to $|\mathbf{m}_0|$ do
$r \leftarrow \mathsf{H.Ev}(hk, \mathbf{m}_b[i], 1^{\mathsf{LKE.il}}); (\mathbf{C}_1[i], tK) \leftarrow \mathsf{LKE.Enc.Rg}(pk, r);$
$\mathbf{K}[i] \leftarrow \mathsf{H.Ev}(hk, tK, 1^{\mathsf{LKE.kl}}); \mathbf{C}_2[i] \leftarrow \mathbf{m}_b[i] \oplus \mathbf{K}[i];$
$\mathbf{C} \leftarrow (\mathbf{C}_1, \mathbf{C}_2); b' \xleftarrow{\$} \mathsf{A}_2(\lambda, PK, \mathbf{C});$ Return $(b' \stackrel{?}{=} b)$

If pk is generated by $\mathsf{LKE.Kg}(\cdot)$ then B is simulating Game_0 for A; otherwise B is simulating Game_1. Thus $\Pr[\mathsf{G}^{\mathsf{A}}_0(\cdot)] - \Pr[\mathsf{G}^{\mathsf{A}}_1(\cdot)] \le \mathsf{Adv}^{\mathsf{loss}}_{\mathsf{LKE},\mathsf{B}}(\cdot)$.

Game_2: Replace the hash function $\mathsf{H}(hk, \cdot, \cdot)$ with a random oracle. We construct a UCE adversary (S, D) as follows.

$\mathsf{S}(\lambda)$
$(pk, \perp) \xleftarrow{\$} \mathsf{LKE.Kg}(\lambda); PK \leftarrow pk;$
$b \xleftarrow{\$} \{0,1\}; (\mathbf{m}_0, \mathbf{m}_1) \xleftarrow{\$} \mathsf{A}_1(\lambda);$
For $i = 1$ to $|\mathbf{m}_0|$ do
$r \leftarrow \mathsf{HASH}(\mathbf{m}_b[i], 1^{\mathsf{LKE.il}});$
$(\mathbf{C}_1[i], tK) \leftarrow \mathsf{LKE.Enc.Rg}(pk, r);$
$\mathbf{K}[i] \leftarrow \mathsf{HASH}(tK, 1^{\mathsf{LKE.kl}}); \mathbf{C}_2[i] \xleftarrow{\$} \mathbf{m}_b[i] \oplus \mathbf{K}[i];$
$\mathbf{C} \leftarrow (\mathbf{C}_1, \mathbf{C}_2);$ Return $(b, PK, \mathbf{C})$

$\mathsf{D}(\lambda, hk, L)$
$(b, PK, \mathbf{C}) \leftarrow L;$
$b' \xleftarrow{\$} \mathsf{A}_2(\lambda, PK, \mathbf{C});$
Return $(b' \stackrel{?}{=} b)$

We can see that if HASH is $\mathsf{H.Ev}$, then (S, D) are simulating Game_1, otherwise they are simulating Game_2. Thus, $\Pr[\mathsf{G}_1^{\mathsf{A}}(\cdot)] - \Pr[\mathsf{G}_2^{\mathsf{A}}(\cdot)] \leq \mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H,S,D}}(\cdot)$.

Game_3: identical to Game_2, except that the random oracle now picks a fresh value for every query, regardless of possible repetitions. Now the random oracle in Game_3 is as follows:

$\mathsf{HASH}(x, l)$
$y \xleftarrow{\$} \{0,1\}^l;$ Return y

Let v be a polynomial that bounds $|\mathbf{m}|$. Since the components of $\mathbf{m}$ are distinct, Game_2 and Game_3 are different only if:

1. some tK is repeated due to repeated r, which happens with probability at most $v^2/2^{1+\mathsf{LKE.il}}$.
2. some tK is coincided with $\mathbf{m}_b[i]$ for some i, the probability is bounded by $v^2/2^{\mathsf{LKE.il}}$.

Hence $\Pr[\mathsf{G}_2^{\mathsf{A}}(\cdot)] - \Pr[\mathsf{G}_3^{\mathsf{A}}(\cdot)] \leq 3v^2/2^{1+\mathsf{LKE.il}}$. Finally, $\Pr[\mathsf{G}_3^{\mathsf{A}}(\cdot)] = 1/2$ since the challenge for A_2 is independent of the challenge bit now.

Thus, by summing up there is $\mathsf{Adv}^{\mathsf{priv}}_{\mathsf{DE,A}}(\cdot) \leq 2\mathsf{Adv}^{\mathsf{loss}}_{\mathsf{LKE,B}}(\cdot) + 2\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H,S,D}} + 3v^2/2^{\mathsf{LKE.il}}$.

Now we should prove the statistical unpredictability of S. The leakage of S is $L = (b, PK, \mathbf{C})$. Let P be a statistical predictor with maximum output size q, and the task of P is finding any $\mathbf{m}_b[i]$ or intermediate value tK. In the original unpredictability game, S is interacting with a normal random oracle. However, if we replace the random oracle with the one defined in Game_3, then L contains no information of $\mathbf{m}_b$ or any tK. Thus, the guessing probability as to $\mathbf{m}_b$ is bounded by $qv\mathsf{Guess}_A(\cdot)$, and the guessing probability as to tK is bounded by $qv/2^{\delta}$, where δ is the lossiness of LKE. By summing up, there is $\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{S,P}}(\cdot) \leq 3v^2/2^{1+\mathsf{LKE.il}} + qv\mathsf{Guess}_{\mathsf{A}}(\cdot) + qv/2^{\delta}$, which is negligible. □

Let the lossy KEM be a "LTDF + UCE" combination, then we get the DPKE scheme in [3] as a special case. However, if we construct the lossy KEM with

"LTDR + UCE" or "entropic projective hash + UCE" then we can get better efficiency with the same security, since generally LTDR is considered to be more efficient than LTDF, as stated in [25].

7 Conclusion

In this paper, we abstract the KEM usage of several lossy primitives and introduce a new lossy primitive lossy KEM. Lossy KEM can be constructed from previous lossy primitives such as LTDF and LTDR, and from entropic projective hashing. With lossy KEM, we generalize previous constructions of lossy encryption and DPKE, and get better efficiency.

Acknowledgments. We are grateful to anonymous reviewers for their helpful comments. The authors are supported by the National Natural Science Foundation of China. Specifically, Yamin Liu is supported by No. 61502480, Xianhui Lu is supported by No. 61572495 and No. 61272534, Bao Li is supported by No. 61379137, and Haiyang Xue is supported by No. 61602473.

References

1. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and efficiently searchable encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007). doi:10.1007/978-3-540-74143-5_30
2. Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic encryption: definitional equivalences and constructions without random oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 360–378. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85174-5_20
3. Bellare, M., Hoang, V.T.: Resisting randomness subversion: fast deterministic and hedged public-key encryption in the standard model. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 627–656. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46803-6_21
4. Bellare, M., Hoang, V.T., Keelveedhi, S.: Instantiating random oracles via UCEs. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 398–415. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40084-1_23
5. Bellare, M., Hoang, V.T., Keelveedhi, S.: Instantiating random oracles via UCEs. In: Cryptology ePrint Archive (2013)
6. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01001-9_1
7. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security, pp. 62–73. ACM (1993)
8. Boldyreva, A., Fehr, S., O'Neill, A.: On notions of security for deterministic encryption, and efficient constructions without random oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85174-5_19

9. Brakerski, Z., Segev, G.: Better security for deterministic public-key encryption: the auxiliary-input setting. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 543–560. Springer, Heidelberg (2011). doi:10.1007/978-3-642-22792-9_31
10. Brzuska, C., Farshim, P., Mittelbach, A.: Indistinguishability obfuscation and UCEs: the case of computationally unpredictable sources. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 188–205. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44371-2_11
11. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998). doi:10.1007/BFb0055717
12. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002). doi:10.1007/3-540-46035-7_4
13. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J. Comput. **33**(1), 167–226 (2003)
14. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24676-3_31
15. Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More constructions of lossy and correlation-secure trapdoor functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 279–295. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13013-7_17
16. Fuller, B., O'Neill, A., Reyzin, L.: A unified approach to deterministic encryption: new constructions and a connection to computational entropy. J. Cryptol. **28**(3), 671–717 (2015)
17. Hemenway, B., Libert, B., Ostrovsky, R., Vergnaud, D.: Lossy encryption: constructions from general assumptions and efficient selective opening chosen ciphertext security. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 70–88. Springer, Heidelberg (2011). doi:10.1007/978-3-642-25385-0_4
18. Kiltz, E., Pietrzak, K., Stam, M., Yung, M.: A new randomness extraction paradigm for hybrid encryption. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 590–609. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01001-9_34
19. Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 18–35. Springer, Heidelberg (2009). doi:10.1007/978-3-642-03356-8_2
20. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. SIAM J. Comput. **40**(6), 1803–1844 (2011)
21. Raghunathan, A., Segev, G., Vadhan, S.: Deterministic public-key encryption for adaptively chosen plaintext distributions. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 93–110. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38348-9_6
22. Wee, H.: Efficient chosen-ciphertext security via extractable hash proofs. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 314–332. Springer, Heidelberg (2010). doi:10.1007/978-3-642-14623-7_17
23. Wee, H.: Dual projective hashing and its applications — lossy trapdoor functions and more. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 246–262. Springer, Heidelberg (2012). doi:10.1007/978-3-642-29011-4_16

24. Xue, H., Li, B., Lu, X., Jia, D., Liu, Y.: Efficient lossy trapdoor functions based on subgroup membership assumptions. In: Abdalla, M., Nita-Rotaru, C., Dahab, R. (eds.) CANS 2013. LNCS, vol. 8257, pp. 235–250. Springer, Heidelberg (2013). doi:10.1007/978-3-319-02937-5_13
25. Xue, H., Lu, X., Li, B., Liu, Y.: Lossy trapdoor relation and its applications to lossy encryption and adaptive trapdoor relation. In: Chow, S.S.M., Liu, J.K., Hui, L.C.K., Yiu, S.M. (eds.) ProvSec 2014. LNCS, vol. 8782, pp. 162–177. Springer, Heidelberg (2014). doi:10.1007/978-3-319-12475-9_12