Regularly Lossy Functions and Their Applications

Yu Chen * Baodong Qin † Haiyang Xue ‡

Abstract

In STOC 2008, Peikert and Waters introduced a powerful primitive called *lossy trapdoor functions* (LTFs). In a nutshell, LTFs are functions that behave in one of two modes. In the normal mode, functions are injective and invertible with a trapdoor. In the lossy mode, functions statistically lose information about their inputs. Moreover, the two modes are computationally indistinguishable. In this work, we put forward a relaxation of LTFs, namely, *regularly lossy functions* (RLFs). Compared to LTFs, the functions in the normal mode are not required to be efficiently invertible or even unnecessary to be injective. Instead, they could also be lossy, but in a regular manner. We also put forward richer abstractions of RLFs, namely *all-but-one regularly lossy functions* (ABO-RLFs) and *one-time regularly lossy filters* (OT-RLFs).

We show that (ABO)-RLFs admit efficient constructions from both a variety of number-theoretic assumptions and hash proof system (HPS) for subset membership problems satisfying natural algebraic properties. Thanks to the relaxations on functionality, the constructions enjoy much compact key size and better computational efficiency than that of (ABO)-LTFs.

We demonstrate the utility of RLFs and their extensions in the leakage-resilient cryptography.

- As a special case of RLFs, lossy functions imply leakage-resilient injective one-way functions with optimal leakage rate $1 - o(1)$.
- ABO-RLFs (or OT-RLFs) immediately imply leakage-resilient one-time message authentication code (MAC) with optimal leakage rate $1 - o(1)$.
- ABO-RLFs together with HPS give rise to leakage-resilient chosen-ciphertext (CCA) secure key encapsulation mechanisms (KEM) (this approach extends naturally to the identity-based setting). Combining the construction of ABO-RLFs from HPS, this gives the first leakage-resilient CCA-secure public-key encryption (PKE) with optimal leakage rate based solely on HPS, and thus goes beyond the barrier posed by Dodis et al. (Asiacrypt 2010).

*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences. School of Cyber Security, University of Chinese Academy of Sciences. Email: chenyu@iie.ac.cn

†National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, P.R. China. Email: baodong.qin@gmail.com

‡State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences. Email: xuehaiyang@iie.ac.cn

Contents

1 Introduction

In STOC 2008, Peikert and Waters [PW08] introduced a powerful primitive called lossy trapdoor function (LTF). Informally, LTF is a collection of functions $\mathcal{F} = \{f_{ek}\}$ whose evaluation key (i.e., function index or code) is created in one of two modes. One is injective (i.e., normal) mode: given a suitable trapdoor td for ek, the entire input x can be efficiently recovered from $f_{ek}(x)$. The other is lossy mode: f_{ek} statistically loses a significant amount of information about its input. Moreover, the two modes are computationally indistinguishable: given just ek, no efficient adversary can tell whether f_{ek} is injective or lossy. They also introduced a richer abstraction called all-but-one lossy trapdoor functions (ABO-LTFs). A collection of ABO-LTFs is associated with a set B called branches. The key generation algorithm takes a given branch $b^* \in B$ as an extra parameter, and outputs an evaluation key ek and a trapdoor td. The function $f_{ek,b}(\cdot)$ is injective and invertible with td for any branch $b \neq b^*$, while the function $f_{ek,b^*}(\cdot)$ is lossy. Moreover, the lossy branch b^* is computationally hidden by ek.

Using LTFs and ABO-LTFs, Peikert and Waters [PW08] develop new approaches for constructing several important cryptographic tools, such as injective TDFs, collision-resistant hash functions (CRHFs), oblivious transfer and CCA-secure PKE.

1.1 Related Work

Since the initial work of [PW08], there has been much additional work on LTFs and related concepts.

One direction of research is to find additional realizations of LTFs. Boyen and Waters [BW10] gave a technique to shrink the public key of matrix construction of [PW08] with the help of pairing. Rosen and Segev [RS09] and Boldyreva et al. [BFO08] independently described simple, compact constructions of LTFs and ABO-LTFs under the decisional composite residuosity (DCR) assumption. Freeman et al. [FGK+13] provided more constructions of LTFs from the quadratic residuosity (QR) and d-linear assumptions. Kiltz et al. [KOS17] and Xue et al. [XLL+13] gave constructions of LTFs based on factoring assumptions. Hemenway and Ostrovsky [HO12] gave a construction of LTFs based on the extended decisional Diffie-Hellman (eDDH) assumption, which generalizes the DDH, QR and DCR assumption. They also showed a generic construction of LTFs from homomorphic smooth HPS. Wee [Wee12] presented an alternative generic construction of LTFs from dual HPS.

Another direction of research is to explore variations and more applications. Rosen and Segev [RS09] and Kiltz et al. [KMO10] showed LTFs imply correlated-product TDFs and adaptive TDFs respectively. Boldyreva et al. [BFO08] constructed CCA-secure deterministic encryption based on LTFs and ABO-LTFs. Hemenway et al. [HLOV11] generalized ABO-LTFs to all-but-N lossy trapdoor functions (ABN-LTFs) that have N lossy branches. Hofheinz [Hof12] further generalized ABN-LTFs to all-but-many (ABM) LTFs in which the number of lossy branches is not bounded by any polynomial. Recently, Boyen and Li [BL17] realized ABM LTFs based on the learning with errors assumptions. So far, ABM-LTFs have shown their usefulness in constructing PKE with strong security properties including selective opening security [Hof12] and key-dependent message security [Hof13]. Mol and Yilek [MY10] constructed a CCA-secure PKE from any slightly lossy trapdoor functions that lose only a noticeable fraction of a bit. On the contrary, Zhandry [Zha16] introduced extremely lossy functions (whose functions in the lossy mode only have polynomial-sized image), and demonstrated extremely lossiness is useful for instantiating random oracles in several settings.

1.2 Motivations

Due to the strong requirement for the normal mode (injective and efficiently invertible with trapdoor), the concrete constructions of (ABO)-LTFs are typically not efficient in terms of the size of evaluation key and complexity of evaluation. The generic constructions of (ABO)-LTFs require advanced property for the basing primitives, such as homomorphic and invertible properties.

In all the known applications of LTFs, the normal mode is used to fulfill functionality, while the lossy mode is used to establish security. However, in many scenarios we do not require the full power of LTFs. As observed by Peikert and Waters [PW08, Section 3.4], some applications (such as injective OWFs, CRHFs) *do not require a trapdoor*, but only indistinguishability between normal mode and lossy mode. Thereby, they conjectured "realizing the weaker notion of lossy (non-trapdoor) functions (LFs) could be achieved more simply or efficiently than the full notion of LTFs", and left the investigation of this question as an interesting problem.

A central goal in cryptography is to base cryptosystems on primitives that are as week as possible. With the question raised by Peikert and Waters [PW08] in mind, we ask the following questions:

> *How to realize LFs efficiently? Are there any other applications of LFs? Can we further weaken the notion of LFs while still being useful?*

1.3 Our Contributions

We answer the above questions affirmatively. An overview of our contributions is as below.

1.4 Regularly Lossy Functions and Extensions

As discussed above, when building cryptographic protocols the normal mode of LTF is used to fulfill functionality. For some applications that invertible property for the normal mode is overkilled, the injective property may also be unnecessary. This suggests that we may further relax the notion of LFs.

We introduce a new primitive called regularly lossy functions (RLFs), which is a public function f_{ek} (the evaluation key ek serves as the function index) that is created to behave in one of two modes. In the normal mode, the function f_{ek} could be lossy, but should lose *regularly* (we will formally define this later). The intuition is that when the input x has high min-entropy, so does $f_{ek}(x)$. In the lossy mode, the function f_{ek} statistically loses a significant amount information about its input x, i.e., the average min-entropy of $x|f_{ek}(x)$ is high. Finally, the two modes are indistinguishable: no efficient adversary can tell whether f_{ek} is in normal mode or lossy mode.

In line of the above intuition, we can use image size to capture the lossy mode same as LTFs [PW08], but not for the normal mode. This is because image size is a *global* characterization for a function, which suffices to give the lower bound of the average min-entropy of $x|f_{ek}(x)$ by applying the chain rule for min-entropy (cf. Lemma A.1), but is insufficient to give the lower bound of the min-entropy of $f_{ek}(x)$. For instance, when the function is highly unstructured, it is possible that the image size of f_{ek} is slightly smaller the domain size, but the min-entropy $f_{ek}(x)$ is much smaller than that of x. To address this subtle issue, we choose a *local* characterization of function named regularity to capture the normal mode. In the normal mode, the function f_{ek} is ν-regular, i.e., each image has at most ν preimages under f_{ek}. With this requirement, the (average) min-entropy of $f(x)$ decreases at most $\log \nu$ compared to that of x (by applying Lemma 2.1 we develop in Section 2.2).

Clearly, our notion of RLFs differs from LFs only at the normal mode, whose functions are not required to be injective but could be flexibly lossy from injective to significantly lossy, subjected to the parameter choices of concrete applications. The only constraint is they should lose in a *regular* way.

To admit more applications, we introduce a richer abstraction called ABO-RLFs, analogously to the extension of LTFs to ABO-LTFs. Briefly, an ABO collection is associated with a branch set B. The generation algorithm of ABO-RLF takes an extra parameter $b^* \in B$, and outputs an evaluation key such that $f_{ek,b}$ is regular for any branch $b \neq b^*$ but is lossy when $b = b^*$. Moreover, the lossy branch is hidden (computationally) by ek. Note that with ABO-RLF, the lossy branch must be determined before publishing ek, which may hinder applications in adaptive scenarios. In line of this limitation, we further introduce an agile version of ABO-RLFs, namely one-time regularly lossy filters (OT-RLFs), in which a lossy branch can be generated on-the-fly even after publishing ek.

1.5 Efficient Constructions of ABO-RLFs

Existing constructions of (ABO)-LTFs are less efficient due to their strong requirement for the normal mode. In contrast, RLFs require nothing but the intrinsic regularity of functions for the normal mode. Such weakening admits much more efficient constructions from both number-theoretic assumptions and HPS.

First, we mainly follow the matrix approach due to [PW08] to give a DDH-based ABO-RLFs, in which the evaluation key is specified by an $n \times m$ matrix over groups. The efficiency improvements of our construction comes from two aspects: (1) since we do not require efficient inversion, the input x can be treated as an n-dimensional vector of elements from some large field (say $\mathbb{Z}_p$) rather than a binary string over $\{0,1\}^n$; (2) since we even do not require injectivity, m could be set smaller than n and thus the matrix size shrinks noticeably. Our DDH-based ABO-RLFs can be naturally extended to base on the eDDH assumption. Second, we give an efficient and direct DCR-based ABO-RLFs, which compares favorably to the DCR-based ABO-LTFs due to [FGK+13] in terms of evaluation key size and computation overhead.

As to generic constructions, we first give a construction of ABO-RLF from any HPS for subset membership problems (SMPs). The construction proceeds via two steps: (1) build LF from any HPS following the approach of building LTF from dual HPS [Wee12]; (2) amplify the obtained RLF to ABO-RLF with branch set $\{0,1\}^\ell$. However, this construction is inefficient in that its second step invokes ℓ individual copies of RLF and involves some degradation in lossiness. Towards a direct and efficient construction, we require the underlying SMP to satisfy natural algebra properties, namely L is a subgroup of X and the quotient group $H = X/L$ is a cyclic group of order p. By exploiting this properties, we manage to give an efficient ABO-RLF with branch set $B = \mathbb{Z}_p$ directly from HPS.

1.6 Applications in Leakage-Resilient Cryptography

On the surface, non-injective function without a trapdoor do not appear pretty useful, since many appealing applications of standard LTF require a trapdoor (e.g., public-key encryption) or at least injectivity (e.g., CRHFs) for the normal mode. Indeed, RLF does not suffice for most of the applications outlined above. Nevertheless, we show that this simple notion on its own or in conjunction with other tools can in fact quite useful in leakage-resilient cryptography.

Traditional security models assume complete privacy of secret keys. However, in real systems the adversary might learn partial information about secret keys by launching various "key leakage attacks" via side channels, which make this idealized assumption false in practice. This

fact leads to the design of leakage-resilient cryptography, which spreads to stream ciphers, block ciphers, digital signatures, public-key encryption, identity-based encryption.

There are several models of key leakage-resilience in the literature, mainly differing in their specifications of what and how many information can be leaked to the adversary. In this work we will focus on a simple yet general model, called bounded-leakage model. In this model, the adversary can learn arbitrary information about the secret key, subjected to the restriction that the total number of leakage is bounded by some leakage bound $\ell(\lambda)$, where λ is the security parameter. The leakage rate is defined as the ratio of $\ell(\lambda)$ to the secret key size $s(\lambda)$, i.e., $\ell(\lambda)/s(\lambda)$. Clearly, $1 - o(1)$ is the optimal leakage rate in the bounded leakage model.

In this work, we demonstrate the utility of RLFs (including their special case – LFs) by exploring their applications in leakage-resilient cryptography.

Leakage-Resilient OWFs. A function is said to be ℓ-leakage-resilient one-way if one-wayness maintains even the attacker may obtain at most ℓ-bits leakage about the preimage.

It was shown in [ADW09b, DHLW10, Kom16] (and implicitly in [ADW09a, KV09]) that any weak universal one-way hash function (UOWHF)[1] from $\{0,1\}^n$ to $\{0,1\}^m$ automatically provides ℓ-leakage-resilient one-wayness, where $\ell \leq n - m - \omega(\lambda)$. The shortcoming of this construction is the resulting LR OWFs are inherently compressing, and the leakage bound is dependent on the image size. As a consequence, in some applications one has to make a trade-off between image size and leakage bound.

In this work, we give an alternative construction based on LF. The insight is that the implication of LF $\Rightarrow$ injective OWF [PW08] also holds in the leakage setting. More precisely, we show that the functions in the injective mode of LFs make up a collection of ℓ-leakage-resilient injective OWFs. The leakage bound is $\ell \leq n - \tau - \omega(\lambda)$, where n is the length of inputs and τ is the logarithm of image size for the lossy mode. Both of our construction based of LF and the construction based on UOWHF achieves optimal leakage rate with appropriate parameter choice. The advantage of our construction is that the leakage bound is independent of the image size[2], which is more applicable in practice. To the best of our knowledge, our construction appears to be the first leakage-resilient injective OWF with optimal leakage rate.

Leakage-Resilient MAC. Hazay et al. [HLAWW13] constructed a leakage-resilient MAC from standard PRF. Though their construction only requires minimum assumption (OWFs), the leakage rate $\log \lambda / s(\lambda)$ is poor. Constructing leakage-resilient MAC under general assmption with higher leakage rate was left as an open problem [HLAWW13].

In this work, we make a progress towards this problem. We construct a leakage-resilient MAC with optimal leakage rate from ABO-RLFs (or OT-RLFs), though in a weaker sense. To convert a ABO-RLF to a MAC, the key generation algorithm generates an evaluation key ek as public parameter, then chooses a random x from input space as the secret key; the tag algorithm treats message m as branch and evaluate $t \leftarrow f_{ek,m}(x)$; the verification algorithm is canonical, namely re-computes the tag and checks for equality.

The resulting MAC turns out to be leakage-resilient strongly unforgeable, though in a weaker sense: the attacker only makes one tagging query and declares the query at the very beginning. The security argument leverages on the power of *lose information*. Upon the attacker submits its target query m^*, the reduction generates ek with m^* as the lossy branch and returns $t^* \leftarrow f_{ek,m^*}(x)$. Observe that f_{ek,m^*} is a lossy function, thus the secret key x still retains sufficient min-entropy even after revealing t^* and bounded leakage. For any forge (m, t), we must have

[1] This is sometimes called second preimage resistant functions.

[2] The leakage bound only subjects to the image size of functions in the lossy mode, which will not be used in real construction.

$m \neq m^*$ since the MAC is unique. Besides, $f_{ek,m}$ is a ν-regular function whenever $m \neq m^*$. In this case, the (average) min-entropy of $t = f_{ek,m}(x)$ decreases at most $\log \nu$ compared to that of x. Therefore, t is unpredictable. The leakage rate could achieve $1 - o(1)$ under proper parameter choice.

The above construction only attains leakage-resilient strong unforgeability in the selective sense, which is not desirable in some scenarios. One could overcome this limitation by relying on OT-RLFs instead of ABO-RLFs, yielding a leakage-resilient one-time strongly unforgeable MAC with optimal leakage rate. We left the construction of full-fledged leakage-resilient MAC as a challenge problem.

Leakage-Resilient PKE. A PKE is said to be ℓ-leakage-resilient if semantic security maintains even if the attacker can obtain at most ℓ-bits leakage about the secret key.

Akavia et al. [AGV09] formalized the notion of leakage-resilient chosen-plaintext security (LR CPA) in the bounded-leakage model. Since then, many existing PKE schemes [Reg05, GPV08, BHHO08] have been proved secure in the bounded-leakage model. Later Naor and Segev [NS09] generalized the main ideas behind these constructions to by giving a generic construction of LR CPA-secure PKE schemes from universal$_1$ hash proof system (HPS) [CS02]. Moreover, they also show how to achieve LR CCA security by either: (1) applying the Naor-Yung paradigm to obtain impractical PKE schemes with leakage-rate $1 - o(1)$ or (2) combining universal$_2$ HPS to obtain practical PKE schemes (variants of the Cramer-Shoup cryptosystems) with leakage-rate $1/6 - o(1)$. Later, Liu et al. [LWZ13] proposed a new variant of the Cramer-Shoup cryptosystems which is LR CCA-secure with leakage-rate $1/4 - o(1)$. Dodis et al. [DHLW10] realized that the HPS approach to building LR CCA-secure PKE seems to be inherently limited to leakage-rates below $1/2$: because the secret-key consists of two components (sk_1 of universal$_1$ HPS for decrypting ciphertext and sk_2 of universal$_2$ HPS for verifying the well-formedness of the ciphertext) and the proofs break down if either of the components is individually leaked in its entirety.[3] Later, Qin and Liu [QL13, QL14] bypassed the bound by replacing the universal$_2$ HPS in the HPS approach [NS09] with a new primitive called one-time lossy filters (OT-LFs). By delicate instantiations of universal$_1$ HPS and OT-LF, they obtained LR CCA-secure PKE schemes with leakage rate $1 - o(1)$. However, if OT-LF is implied by HPS is unknown. The problem of whether we can build LR CCA-secure PKE with optimal leakage-rate based on solely HPS is still open.

In the identity-based setting, Alwen et al. [ADN+10] gave a generic construction of LR CPA-secure IBE based on identity-based HPS. Lewko et al. [LRW11] build LR CPA-secure IBE via dual system encryption. So far, the only known two LR CCA-secure IBE schemes [ADN+10, SGL16] are adapted from the Gentry's IBE [Gen06], with leakage rate $1/6 - o(1)$ and $1/4 - o(1)$ respectively.

In this work, we resolve this problem by building LR CCA-secure PKE with leakage rate $1 - o(1)$ based solely on HPS. This goes beyond previous believed bound conjectured by Dodis et al. [DHLW10]. Our starting point is the work of Qin and Liu [QL13]. It is well-known that key encapsulation mechanism (KEM) is more preferable than PKE from both theoretic and practice interest, thus we focus on the construction of leakage-resilient KEM.

Observe that in the setting of PKE the challenge ciphertext depends on attacker's choice of target messages, whereas in the setting of KEM the challenge ciphertext is entirely deter-

[3] Kiltz et al. [KPSY09] showed that CCA-secure PKE can be constructed from a universal$_2$ HPS with an authenticated one-time secure symmetric encryption, while universal$_2$ HPS can be generically obtained from universal$_1$ HPS via 4-wise independent hash function. At a first glance, their construction can be easily augmented to be leakage-resilient CCA-secure by applying randomness extractor to the projective hash. However, such augment could be very subtle in that the adding of a random seed may render the overall ciphertext easily malleable, and thus cannot be CCA-secure.

mined by the challenger in the setting of KEM. Such feature allows us to replace OT-LFs with all-but-one lossy functions (ABO-LFs), which saves at least a chameleon hash for the KEM construction.[4] Moreover, we show that ABO-LFs can be relaxed to ABO-RLFs. As we show in Section 5, ABO-RLFs can be efficiently constructed from any HPS for subgroup membership problem with natural algebraic properties. Taken together, the secret key in our approach consists of just one component for verifying the well-formedness of the ciphertext and for decrypting it simultaneously. Therefore, the leakage rate of our construction can go beyond the limitation of 1/2, being subject to the leakage tolerance of the underlying universal$_1$ HPS. For instance, applying the DDH-based universal$_1$ HPS from [QL13], we obtain a LR CCA-secure KEM with leakage rate $1/2 - o(1)$; applying the universal$_1$ HPS from refined subgroup indistinguishability problem [QL14], we obtain a LR CCA-secure KEM with leakage rate $1 - o(1)$.

Note that a KEM can be bootstrapped to a PKE by combining a data encapsulation mechanism (DEM) with appropriate security properties [CS02, KD04, HK07], and the composition applies well in the leakage-resilient setting (without requiring DEM to be leakage-resilient). Taken together, our KEM construction indicates that LR-CCA secure PKE with optimal leakage ratio are achievable based on solely HPS.

Notably, our approach also extends to the identity-based setting: one can build LR CCA-secure IB-KEM generically from identity-based HPS and ABO-RLFs (or OT-RLFs), and the leakage rate is decided by the leakage tolerance of the underlying IB-HPS. By combining the IB-HPS realizations from the CPA-secure IBE schemes [Gen06, CDRW10] and appropriate instantiations of ABO-RLFs (or OT-RLFs), we obtain IB-KEMs with leakage rate $1/2 - o(1)$ and $1/3 - o(1)$ respectively, higher than all previous work [ADN+10, SGL16].

We conclude this section by remarking that in (ABO)-RLF' applications to leakage-resilient cryptography (including leakage-resilient MAC, KEM and IB-KEM) presented in this work, the regularity ν allows for an interesting trade-off between efficiency and leakage tolerance, since larger ν typically allows for more efficient realizations of (ABO)-RLFs.

An overview of the contructions of this work is given in Figure 1.

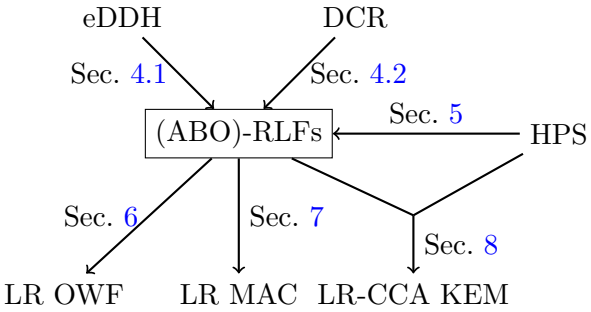

Figure 1: Overview of the results in this work.

2 Preliminaries

2.1 Basic Notations

For a distribution or random variable X, we write $x \xleftarrow{R} X$ to denote the operation of sampling a random x according to X. For a set X, we use $x \xleftarrow{R} X$ to denote the operation of sampling x

[4] As shown in [QL13], OT-LFs can be build from ABO-LFs and chameleon hash.

uniformly at random from X, and use $|X|$ to denote its size. We use U_X to denote the uniform distribution over X.

We denote $\lambda \in \mathbb{N}$ as the security parameter. Unless described otherwise, all quantities are implicit functions of λ, and all cryptographic algorithms (including the adversary) take λ as an input. We say that a quantity is negligible, written $\mathsf{negl}(\lambda)$, if it vanishes faster than the inverse of any polynomial in λ. A probabilistic polynomial time (PPT) algorithm is a randomized algorithm that runs in time $\mathsf{poly}(\lambda)$. If $\mathcal{A}$ is a randomized algorithm, we write $z \leftarrow \mathcal{A}(x_1, \ldots, x_n; r)$ to indicate that $\mathcal{A}$ outputs z on inputs $(x_1, \ldots, x_n)$ and random coins r. For notational clarity we usually omit r and write $z \leftarrow \mathcal{A}(x_1, \ldots, x_n)$.

Due to space limit, we defer the definition of standard cryptographic primitives and information background to Appendix A.

2.2 Regular Functions

A function f is injective (akin, 1-to-1) if every image has one and only one preimage. Following [BHSV98], we measure the amount of "non-injectivity" by looking at the maximum preimage size. Let ν be a quantity of security parameter λ. We say that f is ν-to-1 (or ν-approximately-regular) if ν bounds the maximum preimage size of f: any image has at most ν preimages under f. Particularly, if every image has the same number (say ν) of preimages, we say f is ν-regular.

We develop the following technical lemma which establishes the relation between the min-entropy of X and $f(X)$.

Lemma 2.1. *Let $f : D \to R$ is a ν-to-1 function and X is a random variable over domain D. Then we have:*

$$\mathsf{H}_\infty(f(X)) \geq \mathsf{H}_\infty(X) - \log \nu$$

Proof. Let x^* be the value in the domain that maximizes $\Pr[X = x]$ and y^* be the value in the range that maximizes $\Pr[f(X) = y]$. Since every image has at most ν preimages, it follows that $\Pr[f(X) = y^*] = \sum_{x \in f^{-1}(y^*)} \Pr[X = x] \leq \nu \cdot \Pr[X = x^*]$. According to the definition of min-entropy, the lemma immediately follows. The equality achieves when f is ν-regular and X follows the uniform distribution. Moreover, the above relation applies to average min-entropy as well. Suppose X is correlated to another random variable Y, we have $\tilde{\mathsf{H}}_\infty(f(X)|Y) \geq \tilde{\mathsf{H}}_\infty(X|Y) - \log \nu$. □

Hereafter, we do not distinguish ν-approximately-regular and ν-regular. For ease of presentation, we refer to them collectively as ν-regular.

3 Regularly Lossy Functions and Extensions

3.1 Regularly Lossy Functions

Now, we define the notion of RLFs. Suppose the size of domain is $2^{n(\lambda)}$ where $n(\lambda) = \mathsf{poly}(\lambda)$. Define $\nu(\lambda) \leq 2^{n(\lambda)}$ to represent the *non-injectivity* of the collection, and $2^{\tau}(\lambda) \leq 2^{n(\lambda)}$ to represent the *image size* of the collection. For all these quantities, we often omit the dependence on the security parameter λ.

A collection of (ν, τ)-RLFs is given by four polynomial time algorithms satisfying the following properties:

- $\mathsf{Setup}(\lambda)$: on input λ, output public parameter pp which includes the descriptions of evaluation key space EK, domain X and range Y.

- $\mathsf{GenNormal}(pp)$: on input pp, output an evaluation key ek. $f_{ek}(\cdot)$ is a ν-regular function from X to Y.
- $\mathsf{GenLossy}(pp)$: on input pp, output an evaluation key ek. $f_{ek}(\cdot)$ is a lossy function from X to Y whose image has size at most 2^{τ}. The *lossiness* is defined as $n - \tau$.
- $\mathsf{Eval}(ek, x)$: on input ek and an element $x \in X$, output $y \leftarrow f_{ek}(x)$.

Hard to distinguish normal from lossy. For all $pp \leftarrow \mathsf{Setup}(\lambda)$, the outputs of $\mathsf{GenNormal}(pp)$ and $\mathsf{GenLossy}(pp)$ are computationally indistinguishable.

Remark 3.1. Our notion of RLFs is a generalization of LFs. In the case $\nu = 1$, RLFs obviously boil down to LFs.

3.2 All-But-One Regularly Lossy Functions

To admit more applications, it is convenient to work with a richer notion named ABO-RLFs. The extension is an analog of LTFs to ABO-LTFs in [PW08]. In an ABO collection, each function has an extra input called its *branch*. All of the branches are regular functions, except for one branch is lossy. The lossy branch is an auxiliary input to the evaluation key generation algorithm, and its value is hidden (computationally) by the resulting evaluation key.

We retain the same notation for n, ν, τ as above, and let B be the set of branches. A collection of (ν, τ)-ABO-RLFs consists of three polynomial time algorithms satisfying the following properties:

- $\mathsf{Setup}(\lambda)$: on input λ, output public parameter pp which specifies of evaluation key space EK, branch set B, domain X and range Y.
- $\mathsf{Gen}(pp, b^*)$: on input pp and any $b^* \in B$, output an evaluation key ek. For any $b \neq b^*$, $f_{ek,b}(\cdot)$ is a ν-regular function from X to Y, while $f_{ek,b^*}(\cdot)$ is a lossy function from X to Y whose image has size at most 2^{τ}.
- $\mathsf{Eval}(ek, b, x)$: on input an evaluation key ek and a branch $b \in B$ and an element $x \in X$, output $y \leftarrow f_{ek,b}(x)$.

Hidden lossy branch. For any $b_0^*, b_1^* \in B \times B$, the output ek_0 of $\mathsf{Gen}(pp, b_0^*)$ and the output ek_1 of $\mathsf{Gen}(pp, b_1^*)$ are computationally indistinguishable.

3.3 One-Time Regularly Lossy Filter

For ABO-LTFs, the lossy branch is fixed as soon as ek is published. This stipulates that the reduction must determine a lossy branch at the very beginning, and thus potentially hinders applications in adaptive scenarios. Qin and Liu [QL13] introduced the notion of OT-LFs, in which a lossy branch could be generated on-the-fly in a somewhat semi-customized (or adversary-dependent) manner. In this work, we generalize OT-LFs to one-time regularly lossy filters (OT-RLFs), which could be thought as an agile version of ABO-RLFs.

A collection of (ν, τ)-OT-RLFs consists of four polynomial time algorithms satisfying the following properties:

- $\mathsf{Setup}(\lambda)$: on input λ, output public parameter pp which includes the descriptions of evaluation key space EK, branch set $B = B_c \times B_a$ (where B_c is the core branch set and B_a is the auxiliary branch set), domain X and range Y.
- $\mathsf{Gen}(pp)$: on input pp, output an evaluation key ek and a trapdoor td. B contains two disjoint subsets, the subset of regular branches B_{normal} and the subset of lossy branches B_{lossy}. For any $b \in B_{\text{normal}}$, $f_{ek,b}(\cdot)$ determines a ν-regular function from X to Y. For any

$b \in B_{\text{lossy}}$, $f_{ek,b}(\cdot)$ determines a lossy function from X to Y whose image has size at most 2^τ.

- $\mathsf{SampLossy}(td, b_a)$: on input a trapdoor td and an auxiliary branch b_a, output a core branch b_c such that $b = (b_c, b_a)$ is lossy branch from B_{lossy}.
- $\mathsf{Eval}(ek, b, x)$: on input ek, $b \in B$ and an element $x \in X$, output $y \leftarrow f_{ek,b}(x)$.

Indistinguishability. For any auxiliary branch $b_a \in B_a$, a lossy core branch $b_c \leftarrow \mathsf{SampLossy}(td, b_a)$ and a random core branch $b_c \xleftarrow{R} B_c$ are computationally indistinguishable.

Evasiveness. For any PPT adversary, it is hard to generate a new lossy branch even given a lossy branch.

3.4 Basic Relations

Peikert and Waters [PW08] showed that LTFs and ABO-LTFs are equivalent for appropriate choices of parameters and degree of lossiness. It is straightforward to verify the equivalence also holds in our regularly lossy setting. We list the results as below for completeness. The security proofs are omitted here since they follow readily from [PW08].

Lemma 3.1. *There exists a collection of (ν, τ)-ABO-RLFs having exactly two branches if and only if there exists a collection of (ν, τ)-RLFs.*

Lemma 3.2. *If there exists a collection of (ν, τ)-ABO-RLFs with branch set $\{0,1\}$, then for any $\ell \geq 1$ there exists a collection of $(\nu, \ell\tau)$-ABO-RLFs with branch set $B = \{0,1\}^\ell$.*

Qin and Liu [QL14] showed that OT-LFs can be generically constructed from ABO lossy functions and chameleon hash functions. The same construction applies to the regularly lossy setting as well.

Lemma 3.3. *If there exists a collection of (ν, τ)-ABO-RLFs and a chameleon hash function, then there exists a collection of (ν, τ)-OT-RLFs.*

4 Concrete Construction of ABO-RLFs

In this section, we build ABO-RLFs from the DDH and DCR assumptions (cf. definition in Appendix B).

4.1 A DDH-based ABO-RLFs

Our construction mainly follow the matrix approach due to [PW08], but with important refinement for better efficiency.

We first recall the algorithm named $\mathsf{GenConceal}$ for generating a pseudorandom concealer matrix that enjoys certain useful linearity properties from [PW08]. In a nutshell, $\mathsf{GenConceal}$ takes as input positive integers n and m (where $n \geq m$), outputs a $n \times m$ matrix $\mathbb{G}^{n \times m}$, in which the matrix is pseudorandom and all the columns lie in a one-dimensional subspace. More precisely, it works as follows:

- Choose $\mathbf{r} = (r_1, \ldots, r_n) \leftarrow \mathbb{Z}_p^n$ and $\mathbf{s} = (s_1, \ldots, s_m) \leftarrow \mathbb{Z}_p^m$ uniformly at random.
- Let $\mathbf{V} = \mathbf{r} \otimes \mathbf{s} = \mathbf{r}^t\mathbf{s} \in \mathbb{Z}_p^{n \times m}$ be the outer product of $\mathbf{r}$ and $\mathbf{s}$.
- Output $\mathbf{C} = g^{\mathbf{V}} \in \mathbb{G}^{n \times m}$ as the concealer matrix.

Lemma 4.1 ([PW08])**.** *Let $n, m = \mathsf{poly}(\lambda)$. Under the DDH assumption, the conceal matrix $\mathbf{C} = g^{\mathbf{V}} \leftarrow \mathsf{GenConceal}(n, m)$ is pseudorandom over $\mathbb{G}^{n \times m}$.*

Our construction of ABO-RLFs from the DDH assumption is as below.

- $\mathsf{Setup}(\lambda)$: run $(\mathbb{G}, g, p) \leftarrow \mathsf{GroupGen}(\lambda)$, output $pp = (\mathbb{G}, g, p)$ and $B = \mathbb{Z}_p$.
- $\mathsf{Gen}(pp, b^*)$: on input pp and $b^* \in \mathbb{Z}_p$, invoke $\mathsf{GenConceal}(n, m)$ to generate $\mathbf{C} = g^{\mathbf{V}} \in \mathbb{G}^{n\times m}$, output $ek = g^{\mathbf{Y}} = g^{\mathbf{V}-b^*\mathbf{I}'}$, where $\mathbf{I}' \in \mathbb{Z}_p^{n\times m}$, i.e., the ith column is the standard basis vector $\mathbf{e}_i \in \mathbb{Z}_p^n$ for $i \leq n$, and the rest columns are zero vectors.
- $\mathsf{Eval}(ek, b, \mathbf{x})$: on input evaluation key $ek = g^{\mathbf{Y}}$, a branch $b \in \mathbb{Z}_p$ and an element $\mathbf{x} \in \mathbb{Z}_p^n$, output $\mathbf{y} = g^{\mathbf{x}(\mathbf{Y}+b\mathbf{I}')} = g^{\mathbf{x}(\mathbf{V}+(b-b^*)\mathbf{I}')} \in \mathbb{G}^m$.

Lemma 4.2. *Under the DDH assumption, the above construction is a collection of* $(p^{n-m}, \log p)$*-ABO-RLFs for* $n > 1$.

Proof. For any $b \neq b^*$, $(\mathbf{V}, b)$ determines p^{n-m}-to-1 function because the rank of $(\mathbf{Y} + b\mathbf{I}')$ is m and the size of the solution space for every $y \in \mathbb{G}^m$ is p^{n-m}. For $b = b^*$, every output $\mathbf{y}$ is of the form $g^{r'\mathbf{s}}$, where $r' = \mathbf{x}\mathbf{r}^t \in \mathbb{Z}_p$. Because $\mathbf{s}$ is fixed by the function index $\mathbf{V}$, there are at most p distinct outputs of any particular function determined by $(\mathbf{V}, b^*)$. The lossiness is $(n-1)\log p$.

The hidden lossy branch property (under the DDH assumption) follows by an elementary reduction: for any branch $b^* \in \mathbb{Z}_p$ the output of $\mathsf{Gen}(\lambda, b^*)$ is computationally indistinguishable from uniform over $\mathbb{G}^{n\times m}$. □

Remark 4.1. The parameter n controls the size of domain, while the parameter m allows us to manipulate the regularity for the ABO branches in a flexible manner. When $m = n$ the above construction becomes the standard ABO lossy functions because the ABO branches are injective.

In the DDH-based ABO-LTF construction [PW08], the input space is restricted to $\{0,1\}^n$ and m must be larger than n to ensure invertible property. In our construction, we do not require invertible property. Therefore, the input space dramatically extends from $\{0,1\}^n$ to $\mathbb{Z}_p^n$ without expanding the conceal matrix. Moreover, when injective property is not necessary, we could further shrink the matrix by setting m smaller than n. In the matrix-based construction, both the size of evaluation key and the computation cost of evaluation are dominated by n and m. Therefore, compared to the DDH-based ABO-LTFs, our DDH-based ABO-RLFs allows much larger inputs and much better efficiency. The flexible choice of m gives rise to more compact evaluation key. A detailed comparison will be given shortly in Section 4.3.

Following a similar approach due to Hemenway and Ostrovsky [HO12], the above DDH-based construction naturally extends to construction based on the eDDH assumption [HO12], which generalized the DDH, QR and DQR assumptions. Thereby, the above construction also implies a DCR-based ABO-RLFs. Nevertheless, we are able to give a more efficient and direct DCR-based construction, as shown in the next subsection.

4.2 A DCR-based ABO-RLFs

Our direct ABO-RLFs construction from the DCR assumption is as below.

- $\mathsf{Setup}(\lambda)$: run $N \leftarrow \mathsf{GenModulus}(\lambda)$, choose a random $z \in \mathbb{Z}_N$ and compute $g = z^{2N} \bmod N^2$, output $pp = (N, y)$ and set $B = \mathbb{Z}_N$.
- $\mathsf{Gen}(pp, b^*)$: on input pp and a given lossy branch $b^* \in \mathbb{Z}_N$, choose a random r in $\mathbb{Z}_N$, computes and outputs $ek = g^r(1+N)^{-b^*}$.
- $\mathsf{Eval}(ek, b, x)$: on input ek, a branch $b \in \mathbb{Z}_N$ and an element $x \in \{0, \cdots, \lfloor N^2/4 \rfloor\}$, output $y = [ek/(1+N)^b]^x = g^{rx}(1+N)^{(b-b^*)x} \in \mathbb{Z}_{N^2}$.

Lemma 4.3. *Under the DCR assumption, the above construction is a collection of* $(1, \phi(N)/4)$*-ABO-RLFs.*

Proof. For any $b \neq b^*$, $f_{ek,b}$ is an injective function overwhelmingly because g is the generator of $2N$-th residuosity with overwhelming probability. Let ϕ be the Euler phi function. The order of g is at least $\phi(N)/4$, and the order of $g^r(1+N)^{b-b^*}$ is $N\phi(N)/4$ with overwhelming probability. For $b = b^*$, every output g^{rx} is the $2N$-th residuosity. Thus, all the images are less than $\phi(N)/4$, and the lossiness is at least $\log N$.

The hidden lossy branch property follows by the security of Paillier encryption [Pai99] (implied by the DCR assumption): the output of $\mathsf{Gen}(pp, b^*)$ is actually a Paillier encryption of b^* with randomness r. Therefore, for any $b_0^*, b_1^* \in \mathbb{Z}_N$, the outputs of $\mathsf{Gen}(pp, b_0^*)$ and $\mathsf{Gen}(pp, b_1^*)$ are computationally indistinguishable.

Since the regularity is 1, this construction is actually a ABO-LF. □

4.3 Efficiency Comparison

In this section, we compare our constructions of ABO-RLFs and previous constructions of ABO-LTFs. The results are summarized in Table 4.3.

Table 1: ABO-LTFs vs. ABO-(R)LFs based on concrete assumptions

ABO-LTF/(R)LF	Assump.	Input size	Lossiness	Key size	Efficiency
ABO-LTF [PW08]	DDH	n	$n - \log p$	$nm\lvert\mathbb{G}\rvert$	nm Add
ABO-RLF Sec. 4.1	DDH	$n \log p$	$(n-1)\log p$	$nm\lvert\mathbb{G}\rvert$	nm (Exp+Add)
ABO-LTF [FGK+13]	DCR	$(s-1)\log N$	$(s-2)\log N$	$\lvert\mathbb{Z}_{N^s}^*\rvert$	1 Exp
ABO-LF Sec. 4.2	DCR	$2\log N - 2$	$\log N$	$\lvert\mathbb{Z}_{N^2}^*\rvert$	1 Exp

Add and Exp denote an addictive and an exponential operation in the underlying group respectively.

By setting $n = m = 2\log p$ in the first line and setting $n = 2, m = 1$ in the second line, our DDH-based ABO-RLF in Section 4.1 has the same input size and lossiness as the DDH-based ABO-LTF [PW08], but the evaluation key size as well as computation overhead is much smaller. By setting $s = 3$ in the third line, our DCR-based ABO-LF in Section 4.2 has roughly the same input size and lossiness as that of the DCR-based ABO-LTF [FGK+13], but the evaluation key as well as computation overhead is much smaller.

5 Generic Construction of ABO-RLFs

In this section, we focus on generic construction of ABO-RLFs.

5.1 Construction from HPS for Subset Membership Problem

Lemma 3.1 and 3.2 indicate that ABO-RLF is implied by RLF. Thus, the task of constructing ABO-RLF can be reduced to seeking generic construction of RLF.

Wee [Wee12] introduced the notion of dual HPS. As with universal HPS, dual HPS also centers around a family of hash function $\{\Lambda_{sk}\}$ indexed by secret key sk and whose input x comes from some "hard" language. As before, dual HPS requires that for $x \in L$ (YES instance), the hash value $\Lambda_{sk}(x)$ is completely determined by x and $pk = \alpha(sk)$. On the other hand, for $x \notin L$ (NO instance), dual HPS requires *invertibility* – that $\alpha(sk)$ and $\Lambda_{sk}(x)$ jointly determine sk, and there exists an inversion trapdoor td that enables us to efficiently recover sk given $(\alpha(sk), \Lambda_{sk}(x))$[5] along with x. Wee showed an elegant construction of LTF from dual HPS,

[5]Following the treatment of [Wee12], we will write $\Lambda_{sk}(x)$ as $\Lambda_x(sk)$ occasionally.

which is depicted in Equation (1) as below.

$$f_x(sk) = \alpha(sk)||\Lambda_x(sk) \tag{1}$$

In Wee's construction, instance x serves as the evaluation key and secret key sk acts as input. The injective mode (when $x \notin L$) follows from the invertible property of dual HPS, whereas the lossy mode (when $x \in L$) follows from the projective property of $\Lambda_{sk}(\cdot)$. Moreover, the indistinguishability of injective and lossy mode follows from the hardness of subset membership problem.

Interestingly, we can build RLF from any HPS via the same construction shown as above. Since RLF is much weaker then LTF, we only need the projective property of HPS; any additional properties such as smooth, universal or invertible properties are unnecessary. Formally, let $(X, L, W, \mathsf{R}, PK, SK, \alpha, \Pi, \Lambda)$ be public parameter of HPS. Assume $f_x(sk) = \alpha(sk)||\Lambda_x(sk)$ is a ν-to-1 function from SK to Π for any $x \notin L$.[6] We have the following lemma.

Lemma 5.1. *Under the subset membership assumption, Equation (1) yields a collection of $(\nu, \log|\mathrm{Img}(\alpha)|)$-RLFs.*

Proof. Correctness for the normal mode follows readily from the fact that $f_x(\cdot)$ is a ν-to-1 function. Lossiness for the lossy mode follows readily from the projective property, which implies that for any $x \in L$, $\mathrm{Img}(f_x) = \mathrm{Img}(\alpha)$. The indistinguishability between normal mode and lossy mode can be directly reduced to the subset membership assumption. □

Putting all the above together, we can generically construct ABO-RLF from any HPS. The construction proceeds via two steps: (1) build RLF from any HPS; (2) amplify the obtained RLF to ABO-RLF with branch set $\{0,1\}^\ell$. However, this generic construction is not efficient in that its second step invokes ℓ individual copies of RLF and involves some degradation in lossiness.

5.2 Efficient Construction from HPS for Algebraic Subset Membership Problems

The above construction serves as a proof of concept that one can generically build ABO-RLF from any HPS. It is intriguing to know if there exists more efficient construction.

Our idea is to exploit more algebra property of the associated subset membership problem. More precisely, we choose to work with group-oriented SMPs, which we call algebraic subgroup membership problem.

Algebraic subset membership problems. We first formally introduce a new class of cryptographic indistinguishability problem called algebraic subset membership problems (ASMPs), which is a special type of SMPs (cf. definition in Section 8) with the following requirements.

1. X forms a finite Abelian group, L forms a subgroup of X.
2. The quotient group $H = X/L$ is cyclic with order $p = |X|/|L|$.

With the above algebraic properties, we have the following two useful facts:

- Let $\overline{a} = aL$ for some $a \in X \backslash L$ be a generator of H, then the co-sets $(aL, 2aL, \ldots, (p-1)aL, paL = L)$ constitute a partition of X.
- For each $x \in L$, $ia + x \in X \backslash L$ for $1 \leq i < p$.

[6] The regularity of α gives an upper bound of ν.

The hardness of ASMPs is same as that of SMPs, which stipulates the uniform distributions over L and $X\backslash L$ are computationally indistinguishable. Define the density of L as $\rho = |L|/|X|$. When ρ is negligible, $U_L \approx_c U_{X\backslash L}$ is equivalent to $U_L \approx_c U_X$ in that $U_{X\backslash L}$ and U_X are statistically close. When ρ is known, $U_L \approx_c U_{X\backslash L}$ implies $U_L \approx_c U_X$ since one can efficiently reconstruct U_X from U_L, $U_{X\backslash L}$ and ρ.

To demonstrate the generality of ASMP, we instantiate it based the DDH, d-linear, QR and DCR assumptions respectively. Due to space limit, we defer the instantiations to Appendix B.

Remark 5.1. ASMP could also be thought as an enhancement of subgroup membership problems with requirement (2). For our application in this work, requirement (2) could be further relaxed to H contains a cyclic subgroup.

Comparison to (refined) subgroup indistinguishability problems. Brakerski and Goldwasser [BG10] introduced the so called subgroup indistinguishability problems (SIPs). SIPs is also defined w.r.t. a finite Abelian group X and a subgroup L. In addition, SIPs require X is isomorphic to direct product of two groups: $X \simeq L \times M$ and $\mathsf{gcd}(\mathrm{ord}(L), \mathrm{ord}(M)) = 1$. Qin and Liu [QL14] introduced refined SIPs, which further requires M to be cyclic. Compared to (refined) SIPs, ASMPs only require the quotient group X/L to be cyclic. Therefore, ASMP is strictly stronger than RSIP, and also arguably stronger than SIP because SIP is unlikely to be implied by the DDH and d-linear problems. Correspondingly, our algebraic subset membership assumption is potentially weaker.

Now we are ready to construct ABO-RLF from HPS for ASMP.

- $\mathsf{Setup}(\lambda)$: run $\mathrm{HPS.}\mathsf{Setup}(\lambda)$ to generate $pp = (X, L, W, \mathsf{R}, PK, SK, \alpha, \Pi, \Lambda)$, pick a random generator aL for the quotient group H, output $\hat{pp} = (pp, a)$.
- $\mathsf{Gen}(\hat{pp}, b^*)$: on input $\hat{pp} = (pp, a)$ and a given lossy branch $b^* \in \mathbb{Z}_p$, run $(x, w) \leftarrow \mathrm{HPS.}\mathsf{SampYes}(pp)$ to sample a random element from L, compute the evaluation key $ek = -b^*a + x \in X$.
- $\mathsf{Eval}(ek, b, sk)$: on input an evaluation key $ek = -b^*a + x$, a branch b and an input sk, compute $\alpha(sk)||\Lambda_{sk}(ek + ba)$. This algorithm defines $f_{ek,b}(sk) := \alpha(sk)||\Lambda_{sk}(ek + ba)$.

Theorem 5.2. *Assume $X = \{0,1\}^n$ and the function $f_x(sk) = \alpha(sk)||\Lambda_x(sk)$ is a ν-regular for any $x \notin L$. The above construction yields a collection of $(\nu, \log|\mathrm{Img}\alpha|)$-ABO-RLFs under the algebraic subset membership problem.*

Proof. By the group property of the ASMP, $ek + ba = x + (b - b^*)a \notin L$ as long as $b \neq b^*$. In this case, $f_{ek,b}(\cdot)$ is a ν-regular function. When $b = b^*$, $ek + ba = x + (b - b^*)a = x \in L$. In this case, $f_{ek,b}(\cdot)$ is a lossy function by the projective property. For the security, the hidden lossy branch property follows readily from the subgroup membership problem. For any $b_0^*, b_1^* \in \mathbb{Z}_p$, $(-b_0^*a + x) \equiv_c (-b_0^*a + u) \equiv u \equiv (-b_1^*a + u) \equiv_c (-b_1^*a + x)$, where $u \xleftarrow{\mathrm{R}} X$. This proves the theorem. □

6 Leakage-Resilient One-Way Functions

We now show LFs implies a family of leakage-resilient OWFs. The construction and security proof are in the same spirit of the implication LTFs ⇒ injective TDFs given in [PW08]. We prove the implication also holds in the leakage setting.

Theorem 6.1. *Suppose* $(\mathsf{Setup}, \mathsf{GenInj}, \mathsf{GenLossy}, \mathsf{Eval})$ *give a collection of lossy functions over* $\{0,1\}^n$ *whose the image size of functions in the lossy mode is at most* 2^τ. *Then* $(\mathsf{Setup}, \mathsf{GenInj}, \mathsf{Eval})$ *is a collection of ℓ-leakage-resilient injective OWFs over* $\{0,1\}^n$ *for any* $\ell \leq n - \tau - \omega(\lambda)$.

Proof. We proceed via two games. Let S_i be the event that $\mathcal{A}$ wins in Game i.

Game 0. This is the standard leakage-resilient game for injective OWFs. $\mathcal{CH}$ interacts with $\mathcal{A}$ as below:

1. Setup: $\mathcal{CH}$ generates $ek \leftarrow \mathsf{GenInj}(\lambda)$, picks $x^* \xleftarrow{R} \{0,1\}^n$, computes $y^* \leftarrow f_{ek}(x^*)$, then sends (ek, y^*) to $\mathcal{A}$ as the challenge.
2. Leakage Queries: $\mathcal{A}$ may adaptively make leakage queries. For each leakage query $\langle g \rangle$, $\mathcal{CH}$ responds with $g(x^*)$.
3. Invert: $\mathcal{A}$ outputs x and wins if $x = x^*$.

According to the definition, we have:

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = \Pr[S_0]$$

Game 1. The same as Game 1 except in Step 1:

1. Setup: $\mathcal{CH}$ generates $ek \leftarrow \mathsf{GenLossy}(\lambda)$.

By the indistinguishability between injective and lossy mode, we have:

$$|\Pr[S_1] - \Pr[S_0]| \leq \mathsf{negl}(\lambda)$$

It remains to analyze $\Pr[S_1]$ in Game 1. Let ek be any fixed evaluation key generated by $\mathsf{GenLossy}(\lambda)$. Then the probability (over the random choice x^*) that even an unbounded $\mathcal{A}$ predicts x^* is given by the average min-entropy of x^* conditioned on $y^* \leftarrow f_{ek}(x^*)$ and leakage of x^*, i.e., the predictability of x^* given y^* and leakage is at most $2^{-\tilde{\mathsf{H}}_\infty(x^*|(y^*,leak))}$. Because $f_{ek}(\cdot)$ takes at most 2^τ values and the total leakage takes at most 2^ℓ values, Lemma A.1 implies that

$$\tilde{\mathsf{H}}_\infty(x^*|(f_{ek}(x^*), leak)) \geq \mathsf{H}_\infty(x) - \tau - \ell = n - \tau - \ell$$

Because $n-\tau-\ell \geq \omega(\lambda)$, the probability that $\mathcal{A}(ek, y^*, leak)$ outputs x^* is $\mathsf{negl}(\lambda)$. By averaging, the same is true for ek chosen at random by $\mathsf{GenLossy}$. This proves $\Pr[S_1] = \mathsf{negl}(\lambda)$, and so is $\Pr[S_0]$. The completes the proof. □

7 Leakage-Resilient Message Authentication Code

In this section, we construct leakage-resilient MAC from ABO-RLFs and OT-RLFs, respectively.

7.1 Construction from ABO Regularly Lossy Functions

We show how to convert an ABO-RLF to a MAC. The high-level idea is treating input as secret key and branch as message, outputting the function value as tag.

- $\mathsf{Setup}(\lambda)$: run ABORLF.$\mathsf{Setup}(\lambda)$ to generate $pp = (EK, B, X, Y)$ where $|X| = 2^n$ and $B = \{0,1\}^b$, generate $ek \leftarrow$ ABORLF.$\mathsf{Gen}(pp, 0^b)$, output $\hat{pp} = (pp, ek)$. The key space $K = X$, the message space $M = B$ and the tag space $T = Y$.
- $\mathsf{Gen}(\hat{pp})$: pick $k \xleftarrow{R} X$ as the secret key.
- $\mathsf{Tag}(k, m)$: compute $t \leftarrow f_{ek,m}(k)$, output (m, t).
- $\mathsf{Vefy}(k, m, t)$: output 1 if $t = f_{ek,m}(k)$ and 0 otherwise.

Theorem 7.1. *If ABORLF is a collection of (ν, τ)-ABO-RLFs, the above construction is ℓ-leakage-resilient selectively one-time sUF as long as $\omega(\log \lambda) \leq n - \tau - \ell - \log \nu$.*

Proof. We proceed via a sequence of games. Let S_i be the event that $\mathcal{A}$ wins in Game i.

Game 0. This is the standard leakage-resilient selective one-time sUF game. $\mathcal{CH}$ interacts with $\mathcal{A}$ as below.

1. Commit and setup: $\mathcal{A}$ declares its single tagging query $\langle m^* \rangle$ before seeing public parameter. $\mathcal{CH}$ generates public parameter by running $pp \leftarrow \text{ABORLF.}\mathsf{Setup}(\lambda)$ and $ek \leftarrow \text{ABORLF.}\mathsf{Gen}(pp, 0^b)$. $\mathcal{CH}$ picks $k \xleftarrow{\text{R}} X$, computes $t^* \leftarrow f_{ek,m^*}(k)$, then sends $\hat{pp} = (pp, ek)$ and t^* to $\mathcal{A}$.
2. Learning phase: $\mathcal{A}$ can make leakage queries adaptively. For each leakage query $\langle g \rangle$, $\mathcal{CH}$ responds with $g(k)$ as long as the total leakage is less than ℓ.
3. Forge: $\mathcal{A}$ outputs (m, t) and wins if $m \neq m^*$ and $t = f_{ek,m}(k)$.

According to the definition, we have:

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = \Pr[S_0]$$

Game 1. The same as Game 0 except that in step 1 $\mathcal{CH}$ generates ek via $\text{ABORLF.}\mathsf{Gen}(pp, m^*)$ rather than $\text{ABORLF.}\mathsf{Gen}(pp, b^*)$. By the hidden lossy branch property, Game 0 and Game 1 are computationally indistinguishable. Therefore, we have:

$$|\Pr[S_1] - \Pr[S_0]| \leq \mathsf{negl}(\lambda)$$

It is left to analyze $\Pr[S_1]$. Since the MAC construction is unique, S_1 is actually the event that $\mathcal{A}$ outputs (m, t) where $m \neq m^*$ and $f_{ek,m}(k) = t$. We now analyze the average min-entropy of t conditioned on $\mathcal{A}$'s view, which is determined by $(pp, ek, m, leak, m^*, t^*)$.

$$\begin{aligned} \tilde{\mathsf{H}}_\infty(t|view) &= \tilde{\mathsf{H}}_\infty(t|ek, m, leak, t^*) && (2) \\ &= \tilde{\mathsf{H}}_\infty(t|ek, m) - \ell - \tau && (3) \\ &\geq n - \log \nu - \ell - \tau && (4) \end{aligned}$$

In the above deduction, Equation (2) follows from the fact that the value $t = f_{ek,m}(k)$ is determined by ek, m and k, while k is independent of m^* and pp. Equation (3) follows from Lemma A.1 and the upper bound of leakage is ℓ-bits and t has at most $2^{n-\tau}$ values. Note that for any $m \neq m^*$, $f_{ek,m}(\cdot)$ is a ν-to-1 function. Combining this fact with Lemma 2.1, Equation (4) immediately follows.

By the parameter choice, we have $n - \log \nu - \ell - \tau \geq \omega(\log \lambda)$. Therefore, $\Pr[S_1] \leq \mathsf{negl}(\lambda)$ holds even against unbounded adversary.

Putting all the above together, the theorem follows. □

7.2 Construction from One-time Regularly Lossy Filters

The above construction based on ABO-RLFs is only provably secure in the selective setting, since the reduction has to program the tagging query m^* as the lossy branch, which must be fixed before publishing the public parameters. Nevertheless, we can circumvent this obstacle by resorting to OT-RLFs. The construction is as below.

- $\mathsf{Setup}(\lambda)$: run $\text{OTRLF.}\mathsf{Setup}(\lambda)$ to generate $pp = (EK, X, Y, B = B_c \times B_a)$ where $|X| = 2^n$, generate $ek \leftarrow \text{OTRLF.}\mathsf{Gen}(pp)$, then output $\hat{pp} = (pp, ek)$. The key space $K = X$, the message space $M = B_a$ and the tag space $T = B_c \times Y$.
- $\mathsf{Gen}(\hat{pp})$: pick $k \xleftarrow{\text{R}} X$ as the secret key.

- $\mathsf{Tag}(k, m)$: set m as the auxiliary branch, pick a random core branch $t_1 \xleftarrow{R} B_c$, compute $t_2 \leftarrow f_{ek,(t_1,m)}(k)$, output $t = (t_1, t_2)$.
- $\mathsf{Vefy}(k, m, t)$: parse $t = (t_1, t_2)$, output 1 if $t_2 = f_{ek,(t_1,m)}(k)$ and 0 otherwise.

Theorem 7.2. *If OTRLF is a collection of (ν, τ)-OT-RLFs, the above construction is ℓ-leakage-resilient one-time sUF as long as $\omega(\log \lambda) \leq n - \tau - \ell - \log \nu$.*

Proof. We proceed via a sequence of games. Let S_i the event that $\mathcal{A}$ wins in Game i.

Game 0. This is the standard leakage-resilient one-time sUF experiment. $\mathcal{CH}$ interacts with $\mathcal{A}$ as below.

1. Setup: $\mathcal{CH}$ generates $pp \leftarrow \text{OTRLF.}\mathsf{Setup}(\lambda)$, then computes $(ek, td) \leftarrow \text{OTRLF.}\mathsf{Gen}(pp)$. $\mathcal{CH}$ sends $\hat{pp} = (pp, ek)$ to $\mathcal{A}$ and keeps td to itself.
2. Learning phase: $\mathcal{A}$ can make leakage queries adaptively and tagging query once. For each leakage query $\langle g \rangle$, $\mathcal{CH}$ responds with $g(k)$ as long as the total leakage is less than ℓ. For the tagging query $\langle m^* \rangle$, $\mathcal{CH}$ picks a random core branch $t_1^* \in B_a$, computes $t_2^* \leftarrow f_{ek,(t_1^*,m^*)}(k)$, and sends $t^* = (t_1^*, t_2^*)$ to $\mathcal{A}$.
3. Forge: $\mathcal{A}$ outputs $(m, t = (t_1, t_2))$ and wins if $t_2 = f_{ek,(t_1,m)}(k)$ and $(m, t) \neq (m^*, t^*)$.

According to the definition, we have:

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = \Pr[S_0]$$

Game 1. The same as Game 0 except that in Step 2 when answering the tagging query m^*, $\mathcal{CH}$ generates by computing $t_1^* \leftarrow \text{OTRLF.}\mathsf{SampLossy}(td, m^*)$ rather than sampling $t_1^* \xleftarrow{R} B_c$. By the indistinguishable property, Game 0 and Game 1 are computationally indistinguishable. Therefore, we have:

$$|\Pr[S_1] - \Pr[S_0]| \leq \mathsf{negl}(\lambda)$$

It is left to analyze $\Pr[S_1]$. S_1 is the event that $\mathcal{A}$ outputs (m, t) where $(m, t) \neq (m^*, t^*)$ and $f_{ek,t_1||m}(k) = t_2$. Since when ek and k are fixed, the branch $t_1||m$ uniquely determines the function value t_2, thus if S_1 happens we must have $(t_1, m) \neq (t_1^*, m^*)$. We now analyze the average min-entropy of t_2 conditioned on $\mathcal{A}$'s view, which is determined by $(pp, ek, m, t_1, leak, m^*, t^*)$.

$$\begin{aligned}
\tilde{\mathsf{H}}_\infty(t_2|view) &= \tilde{\mathsf{H}}_\infty(t_2|ek, m, t_1, leak, t_2^*) && (5)\\
&= \tilde{\mathsf{H}}_\infty(t_2|ek, m) - \ell - \tau && (6)\\
&\geq n - \log \nu - \ell - \tau && (7)
\end{aligned}$$

In the above deduction, Equation (5) follows from the fact that the value $t_2 = f_{ek,(t_1,m)}(k)$ is determined by $ek, (t_1, m)$ and k, while k is independent of m^*, pp and t_1^*. In Game 2, $f_{ek,(t_1^*,m^*)}(\cdot)$ is a lossy function. Equation (6) follows from Lemma A.1 and the upper bound of leakage is ℓ-bits and t_2^* has at most 2^τ values. According to the evasiveness property, $f_{ek,(t_1,m)}(\cdot)$ where $(t_1, m) \neq (t_1^*, m^*)$ is a ν-to-1 function with overwhelming probability. Combining this fact with Lemma 2.1, Equation (7) immediately follows.

By the parameter choice, we have $n - \log \nu - \tau - \ell \geq \omega(\log \lambda)$. Therefore, $\Pr[S_1] \leq \mathsf{negl}(\lambda)$ holds even against unbounded adversary.

Putting all the above together, the theorem follows. □

8 Leakage-Resilient CCA-secure KEM

Our starting point is the work of Qin and Liu [QL13]. By combining a universal HPS and an OT-LF in a clever manner, they obtained a simple and efficient leakage-resilient CCA-secure PKE scheme with higher leakage rate than previous constructions based on HPS [NS09, LWZ13].

To better illustrate our idea, we first briefly review their construction and security proof. Their construction can be divided in two steps. In the first step, they followed the approach of [NS09] to build a LR CPA-secure PKE from a universal$_1$-HPS. The first part of the ciphertext is $(x, s, z = \mathsf{ext}(\pi, s) + m)$, where x is a random element in L with witness w, s is a random seed for randomness extractor ext, m is the message, and $\pi = \mathrm{HPS.Pub}(pk, x, w)$. In the second step, they employed an OT-LF $f_{ek,\cdot}(\cdot)$ to generate a randomized tag to authenticate the first part of the ciphertext. The second part ciphertext is (b_c, t), where b_c is randomly chosen core branch, $x||s||z$ serves as the auxiliary branch b_a, and $t = f_{ek,b_c||b_a}(k)$. This differs from previous (leakage-resilient) CCA-secure PKE constructions which use an independent universal$_2$ HPS to authenticate the first part of the ciphertext, and eventually allows high leakage ratio.

To establish security, the challenge ciphertext $c^* = (x^*, s^*, z^*, b_c^*, t^*)$ evolves via a sequence of hybrids. In the last hybrid, x^* is sampled from $X \backslash L$ and t^* is evaluated via a lossy core branch $b_c^* \leftarrow \mathrm{OTLF.SampLossy}(td, b_a^* = x^*||s^*||z^*)$. No PPT adversary can tell the changes due to the hardness of subset membership problem and the indistinguishability of lossy branches and injective ones. Conditioned on c^*, it is possible that $\pi^* = \mathrm{HPS.Priv}(sk, x^*)$ maintains high min-entropy by proper parameter choice of ext and the fact that t^* is evaluated under a lossy branch. On one hand, when a PPT adversary makes decryption queries, $f_{ek,(b_c,b_a)}(\cdot)$ is an injective function with overwhelming probability due to the evasiveness of OT-LF, and thus the resulting t maintains the min-entropy of its input. According to the universal property of HPS and the fact that t^* is evaluated under a lossy branch, $\Lambda_{sk}(x)$ has high average min-entropy when $x \notin L$ even after exposing c^*. Thereby, the reduction can safely reject all invalid decryption queries with $x \notin L$. On the other hand, due to the projection of Λ_{sk}, the responses to all valid decryption queries do not reveal more information about sk other than pk and c^*. In summary, the decryption oracle does not reveal more information of π^* to the adversary. Upon the this point, ext can be used to distill the leftover entropy from π^* as the session key to mask m.

From both theoretic and practical interest, KEM is more preferable than PKE. In Qin-Liu's PKE, the auxiliary branch b_a is of the from (x, s, z). During the security proof, $z^* = m^* + \mathsf{ext}(\pi^*, s^*)$ cannot be determined by the reduction in advance, in that m^* is one of the two messages outputted by the adversary in the challenge stage. Thereby, the reduction is unable to decide the lossy branch at the very beginning, but has to generate it with the help of trapdoor on-the-fly. In contrast, in the KEM setting the reduction has fully control of the challenge ciphertext $c^* = (x^*, s^*)$, which could be programmed as the lossy branch before the generation of evaluation key. Thereby, the agility of OT-LF is overkilled and its static version – ABO-LF suffices. Moreover, we note that both OT-LF and ABO-LF act as a leakage-resilient MAC in the construction. Combining this observation with the implication we have shown in Section 7, a HPS and an ABO-RLF suffice for the construction of leakage-resilient CCA-secure KEM.

Next, we formally show how to construct leakage-resilient CCA-secure KEM from HPS and ABO-RLF. We first recall the notion of HPS [CS02] as below.

Hash Proof System. A HPS consists of the following algorithms:

- $\mathsf{Setup}(\lambda)$: on input λ, output public parameter $pp = (X, L, W, \mathsf{R}, PK, SK, \alpha, \Pi, \Lambda)$. Here X is a finite non-empty set, L is a proper subset of X defined by binary relation $\mathsf{R} \subset X \times W$ such that $x \in L$ if and only if $(x, w) \in \mathsf{R}$ for some witness $w \in W$. Here PK is the public key space, SK is the secret key space, $\alpha : SK \to PK$ is a projective map, Π is the proof space, $\Lambda = \{\Lambda_{sk} : X \to \Pi\}_{sk \in SK}$ is a family of hash functions indexed by SK.

- $\mathsf{SampYes}(pp)$: on input pp, outputs a random element $x \in L$, together with a witness $w \in W$ for x. We refer to elements belong to L as Yes instances.
- $\mathsf{SampNo}(pp)$: on input pp, output a random element $x \in X \backslash L$. We refer to elements belong to $X \backslash L$ as No instances.
- $\mathsf{KeyGen}(pp)$: on input pp, pick $sk \xleftarrow{\mathrm{R}} SK$, compute $pk \leftarrow \alpha(sk)$, output a key pair (pk, sk).
- $\mathsf{Priv}(sk, x)$: on input sk and $x \in X$, output its hash proof $\pi \leftarrow \Lambda_{sk}(x)$.
- $\mathsf{Pub}(pk, x, w)$: on input pk, $x \in L$ together with a witness w, output $\pi \in \Pi$.

Subset membership problem. Cramer and Shoup [CS02] introduced the subset membership problems (SMP) to abstract natural cryptographic indistinguishability problems such as the DDH and QR problems as well as others.

SMP w.r.t. (X, L, W, R) requires the random distributions over L and $X \backslash L$ are computationally indistinguishable, i.e., for any PPT adversary $\mathcal{A}$, we have:

$$\mathsf{Adv}^{\mathrm{smp}}_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(pp, x_0)] - \Pr[\mathcal{A}(pp, x_1)]| \leq \mathsf{negl}(\lambda)$$

where $pp \leftarrow \mathsf{Gen}(\lambda)$, $(x_0, w) \leftarrow \mathsf{SampYes}(pp)$, and $x_1 \leftarrow \mathsf{SampNo}(pp)$.

Projection. Λ is projective if the action of Λ_{sk} on L is determined by $pk = \alpha(sk)$, i.e., for all $(pk, sk) \leftarrow \mathsf{KeyGen}(pp)$ and all $x \in L$ with witness w, we have:

$$\Lambda_{sk}(x) = \mathsf{Pub}(pk, x, w)$$

Universal$_1$. Λ is ϵ_1-universal$_1$ if for all $pk \in PK$, all $x \in X \backslash L$ and all $\pi \in \Pi$, we have:

$$\Pr[\Lambda_{sk}(x) = \pi | (pk, x)] \leq \epsilon_1$$

where the probability is over all possible sk with $\alpha(sk) = pk$.

The lemma below follows directly from the definition of min-entropy.

Lemma 8.1. *If Λ is ϵ_1-universal$_1$, then for all $pk \in PK$ and $x \in X \backslash L$, $\mathsf{H}_\infty(\Lambda_{sk}(x) | (pk, x)) \geq \log 1/\epsilon_1$, where $sk \leftarrow SK$ with $pk = \alpha(sk)$.*

8.1 Construction from HPS and ABO-RLF

Now, we show how to construct LR CCA-secure KEM from a universal$_1$ HPS, an ABO-RLF and randomness extractor, which is depicted in Figure 2.

- $\mathsf{Setup}(\lambda)$: run $\mathrm{HPS.Setup}(\lambda)$ to generate $pp_1 = (X, L, W, \mathsf{R}, PK, SK, \alpha, \Pi, \Lambda)$[7], where Λ is ϵ_1-universal$_1$ for $n = \log 1/\epsilon_1$; run $\mathrm{ABORLF.Setup}(\lambda)$ to generate $pp_2 = (EK, B = X \times \{0,1\}^d, \Pi, T)$; pick an average-case $(n - \tau - \ell, k, \epsilon_2)$-extractor $\mathsf{ext} : \Pi \times \{0,1\}^d \to K$ where $k = \log |K|$; output $pp = (pp_1, pp_2)$.
- $\mathsf{KeyGen}(pp)$: parse $pp = (pp_1, pp_2)$, then run $(pk, sk) \leftarrow \mathrm{HPS.KeyGen}(pp_1)$ and $ek \leftarrow \mathrm{ABORLF.Gen}(pp_2, 0^{m+d})$, output public key $\hat{pk} = (pk, ek)$ and secret key sk.
- $\mathsf{Encaps}(\hat{pk})$: on input $\hat{pk} = (pk, ek)$, sample $(x, w) \leftarrow \mathrm{HPS.SampYes}(pp_1)$, compute $\pi \leftarrow \mathrm{HPS.Pub}(pk, x, w)$, pick a random seed $s \xleftarrow{\mathrm{R}} \{0,1\}^d$, compute $t \leftarrow f_{ek, x||s}(\pi)$, output $c = (x, s, t)$ and $k \leftarrow \mathsf{ext}(\pi, s)$.
- $\mathsf{Decaps}(sk, c)$: on input sk and $c = (x, s, t)$, compute $\pi \leftarrow \mathrm{HPS.Priv}(sk, x)$, output $k \leftarrow \mathsf{ext}(\pi, s)$ if $t = f_{ek, x||s}(\pi)$ and $\perp$ otherwise.

[7] Assume each element in X can be uniquely encoded as a binary string in $\{0,1\}^m$.

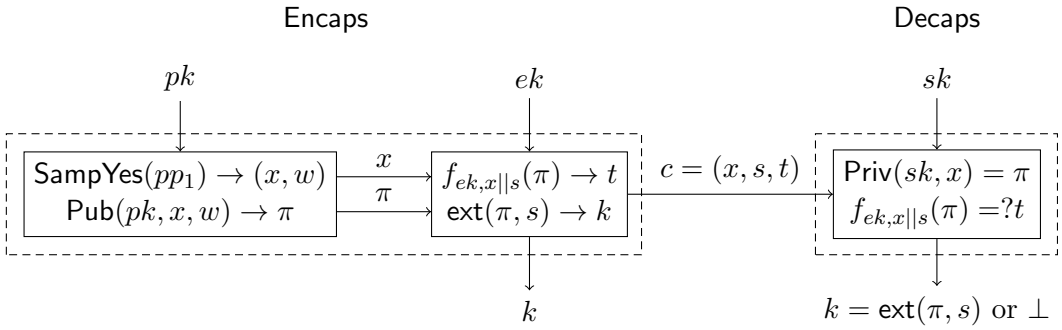

Figure 2: Our approach of KEM construction from HPS and ABO RLF.

Theorem 8.2. *Assuming SMP is hard, HPS is an ϵ_1-universal$_1$ hash proof system, ABORLF is a collection of (ν, τ)-ABO-RLFs and* ext *be an average-case $(n-\tau-\ell, k, \epsilon_2)$-strong extractor, the above construction is ℓ-leakage-resilient CCA-secure as long as $\omega(\log \lambda) \leq n - \tau - \ell - k - \log \nu$.*

Proof. In the following security analysis, we refer to ciphertexts (x, s, t) whose $x \in L$ as valid and $t = f_{ek,x||s}(\pi)$ as well-formed. Clearly, there exist invalid but well-formed ciphertexts.

We proceed via a sequence of games. We start with Game 0, where the challenger $\mathcal{CH}$ proceeds in the standard LR CCA security game (i.e., k_0^* is a real key and k_1^* is a random key) and end up with a game where both k_0^* and k_1^* are chosen uniformly at random. Let S_i be the probability that $\mathcal{A}$ wins in Game i.

Game 0. This is the standard LR CCA security game. $\mathcal{CH}$ interacts with $\mathcal{A}$ as follows:

1. Setup: $\mathcal{CH}$ runs $pp_1 \leftarrow \text{HPS.}\mathsf{Setup}(\lambda)$ and $pp_2 \leftarrow \text{ABORLF.}\mathsf{Setup}(\lambda)$, runs $(pk, sk) \leftarrow \text{HPS.}\mathsf{KeyGen}(pp_1)$, $ek \leftarrow \text{ABORLF.}\mathsf{Gen}(pp_2, 0^{m+d})$, sets sk as secret key and sends $pp = (pp_1, pp_2)$ and $\hat{pk} = (pk, ek)$ to $\mathcal{A}$.
2. Phase 1: $\mathcal{A}$ can make leakage queries adaptively. For each leakage query $\langle g \rangle$, as long as the total leakage is less than ℓ, $\mathcal{CH}$ responds with $g(sk)$.
3. Challenge: $\mathcal{CH}$ samples $\beta \in \{0, 1\}$, $s^* \xleftarrow{\text{R}} S$, $(x^*, w^*) \leftarrow \text{HPS.}\mathsf{SampYes}(pp_1)$, computes $\pi^* \leftarrow \Lambda_{sk}(x^*)$ via $\text{HPS.}\mathsf{Pub}(pk, x^*, w^*)$, $t^* \leftarrow f_{ek,x^*||s^*}(\pi^*)$, $k_0^* \leftarrow \mathsf{ext}(\pi^*, s^*)$, samples $k_1^* \xleftarrow{\text{R}} K$, sends $c^* = (x^*, s^*, t^*)$ and k_β^* to $\mathcal{A}$.
4. Phase 2: $\mathcal{A}$ can make decapsulation queries adaptively. For each decapsulation query $c = (x, s, t)$ where $c \neq c^*$, $\mathcal{CH}$ responds with $\text{KEM.}\mathsf{Decaps}(sk, c)$, that is, computes $\pi \leftarrow \Lambda_{sk}(x)$ via $\text{HPS.}\mathsf{Priv}(sk, x)$, outputs $k \leftarrow \mathsf{ext}(\pi, s)$ if $t = f_{ek,x||s}(\pi)$ and $\perp$ otherwise. The decapsulation query for c^* will be directly rejected.
5. Finally, $\mathcal{A}$ outputs a guess β' for β and wins if $\beta' = \beta$.

According to the definition, we have:

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[S_0] - 1/2|$$

Game 1. Same as Game 0 except that $\mathcal{CH}$ samples (x^*, w^*) and s^* in the setup stage. This change is purely conceptual. Thus, we have:

$$\Pr[S_1] = \Pr[S_0]$$

Game 2. Same as Game 1 except $\mathcal{CH}$ generates $ek \leftarrow \text{ABORLF.}\mathsf{Gen}(pp_2, \cdot)$ with branch $x^*||s^*$ rather than 0^{m+d}. A straightforward reduction to the hidden lossy branch property of ABO-RLF yields:

$$|\Pr[S_2] - \Pr[S_1]| \leq \mathsf{negl}(\lambda)$$

Game 3. Same as Game 2 except that in the challenge stage $\mathcal{CH}$ computes $\pi^* \leftarrow \Lambda_{sk}(x^*)$ via HPS.$\mathsf{Priv}(sk, x^*)$ rather than HPS.$\mathsf{Pub}(pk, x^*, w^*)$. Due to the correctness of HPS, we have:

$$\Pr[S_3] = \Pr[S_2]$$

Game 4. Same as Game 3 except that $\mathcal{CH}$ samples x^* via HPS.SampNo rather than HPS.$\mathsf{SampYes}$. A straightforward reduction to the SMP yields:

$$|\Pr[S_4] - \Pr[S_3]| \leq \mathsf{Adv}^{\mathrm{smp}}_{\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda)$$

Game 5. Same as Game 4 except that $\mathcal{CH}$ directly rejects decapsulation queries $\langle c = (x, s, t)\rangle$ if $x \notin L$.

Let E be the event in Game 5 that $\mathcal{A}$ makes an invalid but well-formed legal decapsulation queries, i.e., $f_{ek,x||s}(\pi) = t$ where $\pi = \Lambda_{sk}(x)$ and $(x, s, t) \neq (x^*, s^*, t^*)$. Clearly, Game 4 and Game 5 are identical if E does not occur. By the difference lemma, we have:

$$|\Pr[S_5] - \Pr[S_4]| \leq \Pr[E]$$

Game 6. Same as Game 5 except that $\mathcal{CH}$ samples $k_0^* \xleftarrow{\mathrm{R}} K$ rather than computing $k_0^* \leftarrow \mathsf{ext}(\pi^*, s^*)$. Clearly, $\mathcal{A}$'s view in Game 6 is identical for either choice of $\beta \in \{0, 1\}$, because β is never used in the experiment. Therefore, we have:

$$\Pr[S_6] = 1/2$$

It remains to analyze $\Pr[E]$ and the relations between Game 5 and Game 6 to establish the main theorem.

Lemma 8.3. $\Pr[E]$ *is negligible in* λ.

Proof. Let E_i be the event that the i-th legal decapsulation query $c = (x, s, t)$ issued by $\mathcal{A}$ is invalid but well-formed. According to the definition of E, we have $E = \cup_{1 \leq i \leq Q_d} E_i$. In what follows, we figure out the upper bounds of $\Pr[E_i]$. Denote by *view* the adversary's view prior to submitting the first decapsulation query. Clearly, $view = (pk, ek, leak, x^*, s^*, t^*, k^*_\beta)$.

Now, to determine the upper bound of $\Pr[E_1]$, we first calculate the average min-entropy of $\Lambda_{sk}(x)|view, x$.

$$\begin{aligned}
\tilde{\mathsf{H}}_\infty(\Lambda_{sk}(x)|view, x) &= \tilde{\mathsf{H}}_\infty(\Lambda_{sk}(x)|pk, x, leak, t^*, k^*_\beta) && (8)\\
&\geq \tilde{\mathsf{H}}_\infty(\Lambda_{sk}(x)|pk, x) - \ell - \tau - k && (9)\\
&= n - \ell - \tau - k && (10)
\end{aligned}$$

In the above deduction, Equation (8) follows from the fact that $\Lambda_{sk}(x)$ is determined by sk and x, while sk is independent of ek, x^* and s^*. Equation (9) follows from Lemma A.1 and the fact that the upper bound of leakage is ℓ-bits, t^* (resp. k^*_β) has at most 2^τ (resp. 2^k) possible values respectively. Equation (10) follows from the ϵ_1-universal$_1$ property of Λ. Note that for a legal and well-formed ciphertext we must have $x||s \neq x^*||s^*$, because the third component of ciphertext is fully determined the first and second components. ek with such branch $x||s \neq x^*||s^*$ determines a ν-regular function. Thereby, the value $t = f_{ek,x||s}(\Lambda_{sk}(x))$ preserves most of the (average) min-entropy of $\Lambda_{sk}(x)$. Combining Lemma 2.1 and Equation (10), $\tilde{\mathsf{H}}_\infty(t|view', x) \geq n - \ell - \tau - k - \log \nu$. This proves that $\Pr[E_1] \leq 2^{\ell+\tau+k+\log\nu}/2^n$. Observe that the adversary can rule out at most ν values of $\Lambda_{sk}(x)$ from each rejection of such invalid decapsulation query, while the responses to all valid decapsulation queries are fully determined by pk and ek (which

do not reveal more information beyond pk), thus $\Pr[E_i] \leq 2^{\ell+\tau+k+\log\nu}/(2^n - i\nu)$. Applying the union bound, we have:

$$\Pr[E] \leq \Sigma_{i=1}^{Q_d} \Pr[E_i] \leq \frac{Q_d 2^{\ell+\tau+k+\log\nu}}{2^n - Q_d\nu} \leq \frac{Q_d}{2^{n-\ell-\tau-k-\log\nu} - Q_d}$$

which is negligible in λ since $n - \tau - \ell - k - \log\nu \geq \omega(\log\lambda)$. This proves the lemma. □

Lemma 8.4. *The adversary's views in Game 5 and 6 are statistically close.*

Proof. We prove this lemma by analyzing the average min-entropy of $\Lambda_{sk}(x^*)$ from the adversary's view. For ease of analysis, we write k^*_β in Game 5 as $k^*_{5,\beta}$ and that in Game 6 as $k^*_{6,\beta}$, and write $leak$ to denote the key leakage. Let $view' = (pk, ek, leak, x^*, s^*, t^*)$. We have:

$$\begin{aligned}
\tilde{\mathsf{H}}_\infty(\Lambda_{sk}(x^*)|view) &= \tilde{\mathsf{H}}_\infty(\Lambda_{sk}(x^*)|pk, x^*, leak, t^*) && (11)\\
&\geq \tilde{\mathsf{H}}_\infty(\Lambda_{sk}(x^*)|pk, x^*) - \ell - \tau && (12)\\
&= n - \ell - \tau && (13)
\end{aligned}$$

In the above deduction, Equation (11) follows from the fact that $\Lambda_{sk}(x^*)$ is independent of ek and s^*. Equation (12) follows from the upper bound of leakage is ℓ and t^* has at most 2^τ possible values. Equation (13) follows from the ϵ_1-universal$_1$ property of the underlying HPS.

Note that $k^*_{5,0} \leftarrow \mathsf{ext}(\Lambda_{sk}(x^*), s^*)$, $k^*_{6,0} \xleftarrow{\mathrm{R}} K$, and ext is an average-case $(n - \tau - \ell, k, \epsilon_2)$-strong extractor, we have $\Delta[(view', k^*_{5,0}), (view', k^*_{6,0})] \leq \epsilon_2$. By the definitions of $k^*_{5,\beta}$ and $k^*_{6,\beta}$, we have $\Delta[(view', k^*_{5,\beta}), (view', k^*_{6,\beta})] \leq \epsilon_2/2$.

Observe that in both Game 5 and Game 6, the responses to all invalid decapsulation queries (whose first element $x \notin L$) are $\perp$, while the responses to all valid decapsulation queries (whose first element $x \in L$) are determined by (pk, ek) according to the projection property of Λ. It follows that the responses to all decapsulation queries in Game 6 are completely determined by a function (possibly inefficient and randomized by the random coins of the adversary), say h, of $(view', k^*_{6,\beta})$, while the responses to all decapsulation queries in Game 6 are completely determined by the same function of $(view', k^*_{6,\beta})$. Denote $\mathcal{A}$'s view in Game 5 by $view_5 = (view', k^*_{5,\beta}, h(view', k^*_{5,\beta}))$ and $\mathcal{A}$'s view in Game 6 by $view_6 = (view', k^*_{6,\beta}, h(view', k^*_{6,\beta}))$. It follows that $\Delta[view_5, view_6] \leq \epsilon_2/2$. This proves the lemma. □

Putting all the above together, the theorem immediately follows. □

Comparison. Compared to Qin-Liu's PKE [QL13, QL14], our construction is more efficient and conceptually simpler. Note that Qin-Liu's PKE requires a universal HPS and an OT-LF, while our construction requires a universal HPS and an ABO-RLF. To date, the only known construction of OT-LF is from ABO-LF and chameleon hash function. As we have shown in Section 4, ABO-RLFs admit more efficient realizations than ABO-LFs. Moreover, as we have show in Section 5, ABO-RLFs can be generically build from any HPS. This implication indicates that our construction can be based solely on HPS, and help us to further reduce the footprint of cryptographic code.

Extension. Our approach for building leakage-resilient CCA-secure KEM naturally extends to the identity-based setting. In the next section, we show how to build leakage-resilient CCA-secure identity-based KEM from identity-based HPS and ABO-RLF (or OT-RLF). The high-level idea is similar but the security proof is more complicated.

9 Leakage-Resilient CCA-secure IB-KEM

In this section, we construct leakage-resilient CCA-secure IB-KEM from IB-HPS and OT-RLF. We first recall the notion of IB-KEM and IB-HPS as below.

9.1 Identity-Based Key Encapsulation Mechanism

An identity-based key encapsulation mechanism (IB-KEM) with identity space I and ciphertext space C and key space K consists of four polynomial time algorithms as follows.

- $\mathsf{Setup}(\lambda)$: on input a security parameter λ, output a master public key mpk and a master secret key msk.
- $\mathsf{Extract}(msk, id)$: on input msk and an identity $id \in I$, output a secret key sk_{id} for id.
- $\mathsf{Encaps}(mpk, id)$: on input mpk and an identity $id \in I$, output a ciphertext $c \in C$ and an encapsulation key $k \in K$.
- $\mathsf{Decaps}(sk_{id}, c)$: on input a secret key sk_{id} and a ciphertext $c \in C$, output a message $k \in K$ or a special reject symbol $\perp$ indicating c is invalid.

Correctness. For all $(mpk, msk) \leftarrow \mathsf{Setup}(\lambda)$ and all $id \in I$ and all $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$ and all $(c, k) \leftarrow \mathsf{Encaps}(mpk, id)$, $\mathsf{Decaps}(sk_{id}, c) = k$ with all but negligible probability over all the randomness in the experiment.

Leakage-Resilient CCA. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary against IBE and define its advantage in the following experiment:

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = \Pr\left[\beta = \beta' : \begin{array}{l} (mpk, msk) \leftarrow \mathsf{Setup}(\lambda); \\ (state, id^*) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{ext}}, \mathcal{O}_{\mathsf{decaps}}(\cdot,\cdot), \mathcal{O}_{\mathsf{leak}}(\cdot)}(mpk); \\ (c^*, k_0^*) \leftarrow \mathsf{Encaps}(mpk, id^*), k_1^* \xleftarrow{\mathrm{R}} K; \\ \beta \xleftarrow{\mathrm{R}} \{0,1\}; \\ \beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{ext}}, \mathcal{O}_{\mathsf{decaps}}(\cdot,\cdot)}(state, c^*, k_\beta^*); \end{array}\right] - \frac{1}{2}.$$

Here $\mathcal{O}_{\mathsf{ext}}(\cdot)$ is an oracle that on input $id \in I$ returns $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$. Note that $\mathcal{O}_{\mathsf{ext}}(\cdot)$ returns the same sk_{id} for repeated extraction queries on id, and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is not allowed to query $\mathcal{O}_{\mathsf{ext}}(\cdot)$ with id^*. $\mathcal{O}_{\mathsf{leak}}(\cdot,\cdot)$ is a leakage oracle that on input $id \in I$ and $g : SK \to \{0,1\}^*$ returns $g(sk_{id})$, and the sum of its output lengths is at most ℓ. $\mathcal{O}_{\mathsf{decaps}}(\cdot,\cdot)$ is a decapsulation oracle that on input $id \in I$ and $c \in C$ where $c \neq c^*$ returns $\mathsf{Decaps}(sk_{id}, c)$. An IB-KEM is said to be ℓ-leakage-resilient CCA-secure if for any PPT adversary $\mathcal{A}$, its advantage defined as above is negligible in λ.

9.2 Identity-Based Hash Proof System

IB-HPS is an extension of HPS in the identity-based setting, which forms the backbone of many IBE schemes including [Gen06, BGH07, Cor09, CDRW10]. We adapt the notion of IB-HPS from [ADN+10] as below.

An IB-HPS consists of the following algorithms:

- $\mathsf{Setup}(\lambda)$: on input λ, output public parameter $pp = (X, L, W, \mathsf{R}, I, SK, \Pi, \Lambda)$ and an associated secret parameter sp. Here X is a finite, non-empty set, I is a set of identities, $L = \{L_{id}\}_{id \in I}$ is a collection of sets defined by a collection of binary relations $\mathsf{R} = \{\mathsf{R}_{id} \subset X \times W\}_{id \in I}$ such that $x \in L_{id}$ if and only if $(x, w) \in \mathsf{R}_{id}$ for some witness $w \in W$. SK is the secret key space, Π is a proof space, and $\Lambda = \{\Lambda_{sk} : X \to \Pi\}_{sk \in SK}$ is a family of hash functions indexed by SK.

- $\mathsf{SampYes}(pp, id)$: on input pp and an identity $id \in I$, output a random element $x \in L_{id}$, together with a witness $w \in W$ for x.
- $\mathsf{SampNo}(pp, id)$: on input pp, output a random element $x \in X \backslash L_{id}$.
- $\mathsf{Extract}(sp, id)$: on input sp and an identity $id \in I$, output a secret key sk_{id}.
- $\mathsf{Priv}(sk_{id}, x)$: on input a secret key sk_{id} and an element $x \in X$, output $\pi \leftarrow \Lambda_{sk_{id}}(x)$.
- $\mathsf{Pub}(id, x, w)$: on input an identity $id \in I$, an element $x \in L_{id}$ together with a witness w, output $\pi \in \Pi$.

Identity-based subset membership problem. The identity-based SMP states that for any $id \in I$ the uniform distributions over L_{id} and $X \backslash L_{id}$ are computationally indistinguishable. Formally, for any PPT adversary $\mathcal{A}$ the following advantage is negligible in λ.

$$\mathrm{Adv}_{\mathcal{A}}(\lambda) = \Pr\left[\beta = \beta' : \begin{array}{l} (pp, sp) \leftarrow \mathsf{Setup}(\lambda); \\ (state, id^*) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{ext}}(\cdot)}(pp); \\ \beta \xleftarrow{\mathrm{R}} \{0,1\}; \\ (x_0^*, w^*) \leftarrow \mathsf{SampYes}(pp, id^*); \\ x_1^* \leftarrow \mathsf{SampNo}(pp, id^*); \\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{ext}}(\cdot)}(state, x_\beta^*); \end{array}\right] - \frac{1}{2}$$

Here $\mathcal{O}_{\mathsf{ext}}(\cdot)$ is an oracle that on input $id \in I$ returns a secret key $sk_{id} \leftarrow \mathsf{Extract}(sp, id)$. We require that $\mathcal{O}_{\mathsf{ext}}(\cdot)$ returns the same secret key for repeated extraction queries on the same identity id.[8] We stress that $\mathcal{A}$ is allowed to query $\mathcal{O}_{\mathsf{ext}}(\cdot)$ with any $id \in I$ (include id^*).

Projection. Λ is projective if the action of $\Lambda_{sk_{id}}$ on L_{id} is determined by id, that is, for all $id \in I$ and all $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$, and for all $x \in L_{id}$ with witness w, we have:

$$\Lambda_{sk_{id}}(x) = \mathsf{Pub}(id, x, w)$$

Universal$_1$. Λ is universal$_1$ if for all $(pp, sp) \leftarrow \mathsf{Setup}(\lambda)$, all $id \in I$ and all $x \in X \backslash L_{id}$, it holds that:

$$\Pr[\Lambda_{sk_{id}}(x))|(pp, sp, id, x)] \leq \epsilon_1$$

where the probability is over all possible $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$.

Lemma 9.1. *If Λ is ϵ_1-universal$_1$, then for all $(pp, sp) \leftarrow \mathrm{IBHPS.Setup}(\lambda)$, all $id \in I$, and all $x \in X \backslash L_{id}$, we have $\mathsf{H}_\infty(\Lambda_{sk_{id}}(x)|(pp, sp, id, x)) \geq \log 1/\epsilon_1$, where $sk_{id} \leftarrow \mathsf{Extract}(sp, id)$.*

9.3 Construction from IB-HPS and OT-RLF

We show how to construct leakage-resilient CCA-secure IB-KEM from an IB-HPS, an OT-RLF and a randomness extractor as below.

- $\mathsf{Setup}(\lambda)$: run $\mathrm{IBHPS.Setup}(\lambda)$ to generate $pp_1 = (X, L, W, \mathsf{R}, I, SK, \Pi, \Lambda)$ and sp_1 where Λ is ϵ_1-universal$_1$ for $n = \log 1/\epsilon_1$; run $\mathrm{OTRLF.Setup}(\lambda)$ to generate $pp_2 = (EK, B = B_c \times B_a, \Pi, T)$ where $B_a = X \times \{0,1\}^d$, compute $ek \leftarrow \mathrm{OTRLF.Gen}(pp_2)$; pick an average-case $(n - \tau - \ell, k, \epsilon_2)$-strong extractor $\mathsf{ext} : \Pi \times \{0,1\}^d \to K$ where $k = \log |K|$, output $mpk = (pp_1, pp_2, ek)$ and $msk = sp_1$.
- $\mathsf{Extract}(msk, id)$: output $sk_{id} \leftarrow \mathrm{IBHPS.Extract}(msk, id)$.

[8] This restriction is natural yet necessary. If an adversary obtains multiple secret keys for the same identity, according to projective and smooth properties of Λ, it can break the language membership problem with high probability by checking if the hash values evaluated under these different secret keys are same.

- $\mathsf{Encaps}(mpk, id)$: on input $mpk = (pp_1, pp_2, ek)$ and an identity $id \in I$, sample $(x, w) \leftarrow \text{IBHPS.}\mathsf{SampYes}(pp_1, id)$, compute $\pi \leftarrow \Lambda_{sk_{id}}(x)$ via $\text{IBHPS.}\mathsf{Pub}(id, x, w)$, pick a random seed $s \xleftarrow{\text{R}} \{0,1\}^d$ and a random core branch $b_c \xleftarrow{\text{R}} B_c$, compute $t \leftarrow f_{ek,(b_c,x||s)}(\pi)$, output a ciphertext $c = (x, s, b_c, t)$ and an encapsulated key $k \leftarrow \mathsf{ext}(\pi, s)$.
- $\mathsf{Decaps}(sk_{id}, c)$: on input a secret key sk_{id} and a ciphertext $c = (x, s, b_c, t)$, compute $\pi \leftarrow \text{IBHPS.}\mathsf{Priv}(sk_{id}, x)$, output $k \leftarrow \mathsf{ext}(\pi, s)$ if $t = f_{ek,(b_c,x||s)}(\pi)$ and $\perp$ otherwise.

Theorem 9.2. *Assuming identity-based SMP is hard, IBHPS is an ϵ_1-universal$_1$ identity-based hash proof system, OTRLF is a collection of (ν, τ)-OT-RLFs and* ext *be an average-case $(n - \tau - \ell, k, \epsilon_2)$-strong extractor, the above construction is leakage-resilient CCA-secure as long as $\omega(\log \lambda) \leq n - \log \nu - \tau - \ell - k$.*

Proof. Our IB-KEM construction follows the same high-level idea behind our KEM construction. However, the security proof turns out to be more complicated. This is because in the CCA-security experiment for IB-KEM the attacker can make decapsulation queries before and after receiving the challenge ciphertext, whereas in the CCA-security experiment for KEM the attacker can only make decapsulation queries after receiving the challenge ciphertext.

We proceed via a sequence of games. We start with Game 0, where the challenger $\mathcal{CH}$ proceeds like in the standard leakage-resilient CCA security game (i.e., k_0^* is a real key and k_1^* is a random key) and end up with a game where both k_0^* and k_1^* are chosen uniformly at random. Let S_i be the probability that $\mathcal{A}$ wins in Game i.

Game 0. This is the standard leakage-resilient CCA security game for IB-KEM. $\mathcal{CH}$ interacts with $\mathcal{A}$ as follows:

1. Setup: $\mathcal{CH}$ generates $(pp_1, sp_1) \leftarrow \text{IBHPS.}\mathsf{Setup}(\lambda)$, $pp_2 \leftarrow \text{OTRLF.}\mathsf{Setup}(\lambda)$, computes $ek \leftarrow \text{OTRLF.}\mathsf{Gen}(pp_2)$, output $mpk = (pp_1, pp_2, ek)$ and $msk = sp_1$.
2. Phase 1: $\mathcal{A}$ can make extraction queries, decapsulation queries and leakage queries adaptively. For each extraction query $\langle id \rangle$, $\mathcal{CH}$ responds with $sk_{id} \leftarrow \text{IBHPS.}\mathsf{Extract}(sp_1, id)$. For each decapsulation query $\langle id, c = (x, s, b_c, t) \rangle$, $\mathcal{CH}$ first computes $\pi \leftarrow \Lambda_{sk_{id}}(x)$ via $\text{IBHPS.}\mathsf{Priv}(sk_{id}, x)$, then outputs $k \leftarrow \mathsf{ext}(\pi, s)$ if $t = f_{ek,(b_c,x||s)}(\pi)$ and $\perp$ otherwise. For each leakage query $\langle g \rangle$, as long as the total leakage is less than ℓ, $\mathcal{CH}$ responds with $g(sk)$.
3. Challenge: $\mathcal{A}$ chooses a target identity id^*. $\mathcal{CH}$ picks $s^* \xleftarrow{\text{R}} \{0,1\}^d$, $b_c^* \xleftarrow{\text{R}} B_c$, samples $(x^*, w^*) \leftarrow \text{IBHPS.}\mathsf{SampYes}(pp_1)$, computes $\pi^* \leftarrow \Lambda_{sk_{id^*}}(x^*)$ via $\text{IBHPS.}\mathsf{Pub}(id^*, x^*, w^*)$, $t^* \leftarrow f_{ek,(b_c^*,x^*||s^*)}(\pi^*)$, $k_0^* \leftarrow \mathsf{ext}(\pi^*, s^*)$. $\mathcal{CH}$ then picks $k_1^* \xleftarrow{\text{R}} K$, $\beta \xleftarrow{\text{R}} \{0,1\}$, sends $c^* = (x^*, s^*, b_c^*, t^*)$ and k_β^* to $\mathcal{A}$.
4. Phase 2: $\mathcal{A}$ can make extraction queries and decapsulation queries. $\mathcal{CH}$ responds the same way as it did in Phase 1 except the extraction query $\langle id^* \rangle$ and decapsulation query $\langle id^*, c^* \rangle$ will be denied.
5. Finally, $\mathcal{A}$ outputs a guess β' for β and wins if $\beta' = \beta$.

According to the definition, we have:

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[S_0] - 1/2|$$

Game 1. Same as Game 0 except that when generating the challenge ciphertext in the challenge stage $\mathcal{CH}$ computes $b_c^* \leftarrow \text{OTRLF.}\mathsf{SampLossy}(td, x^*||s^*)$ rather than $b_c^* \xleftarrow{\text{R}} B_c$. Due to the indistinguishability property of OT-RLF, we have:

$$|\Pr[S_1] - \Pr[S_0]| \leq \mathsf{negl}(\lambda)$$

Game 2. Same as Game 1 except that in the challenge stage $\mathcal{CH}$ computes $\pi^* \leftarrow \Lambda_{sk_{id^*}}(x^*)$ via IBHPS.$\mathsf{Priv}(sk_{id^*}, x^*)$ rather than IBHPS.$\mathsf{Pub}(id^*, x^*, w^*)$. Due to the correctness of IB-HPS, we have:

$$\Pr[S_2] = \Pr[S_1]$$

Game 3. Same as Game 2 except that in the challenge stage $\mathcal{CH}$ samples x^* via IBHPS.SampNo rather than IBHPS.$\mathsf{SampYes}$. Due to the assumed hardness of identity-based SMP, we have:

$$|\Pr[S_3] - \Pr[S_2]| \leq \mathsf{Adv}^{\mathrm{ibsmp}}_{\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda)$$

Game 4. Same as in Game 3 except that in Phase 1 $\mathcal{CH}$ handles decapsulation queries $\langle id, c = (x, s, b_c, t)\rangle$ as follows:

- $x \in L_{id}$: $\mathcal{CH}$ responds normally with sk_{id}.
- $x \notin L_{id}$: If $\mathcal{A}$ had queried the secret key for id, $\mathcal{CH}$ responds with the associated sk_{id}. Else, $\mathcal{CH}$ directly responds with $\perp$.

Let F be the event that $\mathcal{A}$ makes an invalid but well-formed decapsulation query without querying the associated secret key before, i.e., $t = f_{ek,(b_c,x||s)}(\Lambda_{sk_{id}}(x))$ where $x \notin L_{id}$ and $\mathcal{A}$ had not asked secret key for id. According to the definition, Game 4 and Game 5 are identical in $\mathcal{A}$'s view if F never happens. Therefore, we have:

$$|\Pr[S_4] - \Pr[S_3]| \leq \Pr[F]$$

We bound $\Pr[F]$ in Lemma 9.3.

Game 5. Same as in Game 4 except that in Phase 2 $\mathcal{CH}$ handles decapsulation queries $\langle id, c = (x, s, b_c, t)\rangle$ as follows:

- case $id \neq id^*$: $\mathcal{CH}$ handles with associated secret key sk_{id}.
- case $id = id^*$: if $x \in L_{id^*}$, $\mathcal{CH}$ responds with sk_{id^*}; if $x \notin L_{id^*}$, $\mathcal{CH}$ directly rejects.

Note Game 4 and Game 5 only differ at the case $id = id^* \wedge x \notin L_{id^*}$, which can be further split to the sub-cases according to whether $x = x^*$. It is easy to see that the decapsulation queries of the form $\langle id^*, c = (x^*, s^*, b_c^*, t)\rangle$ will be rejected in both Game 4 and Game 5 due to either illegal or ill-formed since the value t is fully determined by the first three components of ciphertext. Let E be the event that $\mathcal{A}$ makes an invalid but well-formed decapsulation queries of id^* with $(b_c, x||s) \neq (b_c^*, x^*||s^*)$. It is easy to see that if E never happens, Game 4 and Game 5 are identical in $\mathcal{A}$'s view. Therefore, we have:

$$|\Pr[S_5] - \Pr[S_4]| \leq \Pr[E]$$

We bound $\Pr[E]$ in Lemma 9.4.

Game 6. Same as Game 5 except that $\mathcal{CH}$ samples $k_0^* \xleftarrow{\mathrm{R}} K$ rather than computing $k_0^* \leftarrow \mathsf{ext}(\pi^*, s^*)$. Clearly, $\mathcal{A}$'s view in Game 6 is identical for either choice of $\beta \in \{0, 1\}$, because β is never used in the experiment. Therefore, we have:

$$\Pr[S_6] = 1/2$$

It remains to analyze $\Pr[F]$, $\Pr[E]$ and the relation between Game 5 and Game 6 to establish the main theorem.

Lemma 9.3. $\Pr[F]$ *is negligible in* λ.

Proof. Let F_{id} be the event that $\mathcal{A}$ makes one such decapsulation query $c = (x, s, b_c, t)$ w.r.t. id. Let $F_{id,i}$ be the event that F_{id} happens in the i-th decapsulation query $c = (x, s, b_c, t)$ of id. According to the definition of F_{id}, we have $F_{id} = \cup_{1 \le i \le Q_d} F_{id,i}$. In what follows, we figure out the upper bounds of $\Pr[F_{id,i}]$. Denote by $view$ the adversary's view prior to submitting the first decapsulation query of id. Let S be the set of identities that $\mathcal{A}$ had asked for secret keys. Note that the responses to decapsulation queries can be expressed as a function h (possibly randomized by $\mathcal{A}$'s random coins) of $(mpk, sk_S, leak)$. This is because the responses to decapsulation queries are fully determined by $\mathcal{A}$'s random coins when $x \in L_{id}$, or by sk_S when $x \notin L_{id}$ (where the responses are either $\mathsf{Decaps}(sk_S, c)$ or $\perp$). Therefore, $view = (mpk, sk_S, leak, h(mpk, sk_S, leak))$.

To determine the upper bound of $\Pr[F_{id,i}]$, we first calculate the average min-entropy of $\Lambda_{sk_{id}}(x)|view, id, x$.

$$\begin{aligned}
\tilde{\mathsf{H}}_\infty(\Lambda_{sk_{id}}(x)|view, id, x) &= \tilde{\mathsf{H}}_\infty(\Lambda_{sk_{id}}(x)|mpk, sk_S, leak, id, x) && (14)\\
&\ge \tilde{\mathsf{H}}_\infty(\Lambda_{sk_{id}}(x)|pp_1, sp_1, leak, id, x) && (15)\\
&\ge \tilde{\mathsf{H}}_\infty(\Lambda_{sk_{id}}(x)|pp_1, sp_1, id, x) - \ell && (16)\\
&= n - \ell && (17)
\end{aligned}$$

In the above deduction, Equation (15) follows from the fact that $\Lambda_{sk_{id}}(x)$ is determined by sk_{id} and x, while sk_{id} is independent of (pp_2, ek), sk_S can be extracted from sp_1 and the random coins used for extraction are independent of sk_{id}. Equation (16) follows from Lemma A.1 and the upper bound of leakage is ℓ-bits. Equation (17) follows from the ϵ_1-universal$_1$ property of Λ. By the evasiveness of OT-RLF, $(b_c, x||s) \in B_{\text{normal}}$ with overwhelming probability, thus the value $t = f_{ek,(b_c,x||s)}(\Lambda_{sk_{id}}(x))$ preserves most of the average min-entropy of $\Lambda_{sk_{id}}(x)$. More precisely, $\tilde{\mathsf{H}}_\infty(t) \ge n - \ell - \log \nu$. Therefore, $\Pr[F_{id,1}] \le 2^{\ell + \log \nu}/2^n$. Observe that $\mathcal{A}$ can rule out ν more value of $\Lambda_{sk_{id}}(x)$ from each rejection of invalid decapsulation query for id, thus $\Pr[F_{id,i}] \le 2^{\ell+\log \nu}/(2^n - i\nu)$. The analysis for other identities are same. Applying the union bound over the decapsulation queries, we have:

$$\Pr[F] \le \frac{Q_d 2^{\ell + \log \nu}}{2^n - Q_d \nu} \le \frac{Q_d}{2^{n - \ell - \log \nu} - Q_d}$$

which is negligible in λ since $n - \ell - \log \nu \ge \omega(\lambda)$. This proves the lemma. □

Lemma 9.4. $\Pr[E]$ *is negligible in* λ.

Proof. Let E_i be the event that the i-th decapsulation query $c = (x, s, b_c, t)$ for id^* made by $\mathcal{A}$ in Phase 2 is invalid but well-formed with $(b_c, x||s) \ne (b_c^*, x^*||s^*)$. Denote by $view$ the adversary's view prior to submitting the first invalid but well-formed decapsulation query of id^* with $(b_c, x||s) \ne (b_c, x^*||s^*)$. Let $view' = (mpk, sk_S, leak, x^*, s^*, b_c^*, t^*, k_\beta^*)$, where sk_S is the set of secret keys for all identities but id^*. According to the definition of Game 5, the responses to all decapsulation queries before can be expressed as a function h (possibly randomized by the random coins of the adversary) of $view'$. Thus, we have $view = (view', h(view'))$.

To determine the upper bound of $\Pr[E_1]$, we first calculate the average min-entropy of $\Lambda_{sk_{id^*}}(x)|view, x, id$.

$$\begin{aligned}
\tilde{\mathsf{H}}_\infty(\Lambda_{sk_{id^*}}(x)|view, x, id) &= \tilde{\mathsf{H}}_\infty(\Lambda_{sk_{id^*}}(x)|view', x, id) && (18)\\
&= \tilde{\mathsf{H}}_\infty(\Lambda_{sk_{id^*}}(x)|pp_1, sk_S, x, id, leak, t^*, k_\beta^*) && (19)\\
&\ge \tilde{\mathsf{H}}_\infty(\Lambda_{sk_{id^*}}(x)|pp_1, sp_1, x, id, leak, t^*, k_\beta^*) && (20)\\
&\ge \tilde{\mathsf{H}}_\infty(\Lambda_{sk_{id^*}}(x)|pp_1, sp_1, x, id) - \ell - \tau - k && (21)\\
&= n - \ell - \tau - k && (22)
\end{aligned}$$

In the above deduction, Equation (18) follows from the fact that h is a function. Equation (19) follows from the fact that sk_{id^*} is independent of pp_2, ek, x^*, s^* and b_c^*. Equation (20) follows from the fact that sk_S can be extracted from sp_1 and the random coins used in the extraction are independent of sk_{id^*}. Equation (21) follows from the upper bound of leakage is ℓ and t^* has at most 2^τ possible values. Equation (22) follows from the ϵ_1-universal$_1$ property of Λ. By the evasiveness of OT-RLF, $(b_c, x||s) \neq (b_c^*, x^*||s^*)$ belongs to B_{normal} with overwhelming probability. Thus, the value $t = f_{ek,(b_c,x||s)}(\Lambda_{sk_{id^*}}(x))$ inherits most of the average min-entropy from $\Lambda_{sk_{id^*}}(x)$. More precisely, $\tilde{\mathsf{H}}_\infty(t) \geq n-\ell-\tau-k-\log\nu$. Therefore, $\Pr[E_1] \leq 2^{\ell+\tau+k+\log\nu}/2^n$. Observe that the adversary can rule out ν more values of $\Lambda_{sk_{id^*}}(x)$ from each rejection of such invalid ciphertext of id^*, we have $\Pr[E_i] \leq 2^{\ell+\tau+k+\log\nu}/(2^n - i\nu)$. Since $\mathcal{A}$ makes at most Q_d decapsulation queries, it follows the union bound that:

$$\Pr[E] \leq \frac{Q_d 2^{\ell+\tau+k+\log\nu}}{2^n - Q_d\nu} \leq \frac{Q_d}{2^{n-\ell-\tau-k-\log\nu} - Q_d}$$

which is negligible in λ since $n-\ell-\tau-k-\log\nu \geq \omega(\log\lambda)$. This proves the lemma. □

Lemma 9.5. *The adversary's views in Game 5 and Game 6 are statistically close.*

Proof. According to the definitions of Game 5 and Game 6, the responses to decapsulation queries in Phase 1 can be expressed as a function h (possibly randomized by $\mathcal{A}$'s random coins) of $(mpk, sk_S, leak)$. Therefore, $\mathcal{A}$'s view in Phase 1 is $view' = (mpk, sk_S, leak, h(mpk, sk_S, leak))$. We now analyze the average min-entropy of $\Lambda_{sk_{id^*}}(x^*)$ conditioned on $view'$, the challenge identity id^*, and the challenge ciphertext $c^* = (x^*, s^*, t^*)$. We have:

$$\begin{aligned}
\tilde{\mathsf{H}}_\infty(\Lambda_{sk_{id^*}}(x^*)|view', c^*, id^*) &= \tilde{\mathsf{H}}_\infty(\Lambda_{sk_{id^*}}(x^*)|pp_1, sk_S, id^*, x^*, t^*) && (23)\\
&\geq \tilde{\mathsf{H}}_\infty(\Lambda_{sk_{id^*}}(x^*)|pp_1, sp_1, id^*, x^*, leak, t^*) && (24)\\
&\geq \tilde{\mathsf{H}}_\infty(\Lambda_{sk_{id^*}}(x^*)|pp_1, sp_1, id^*, x^*) - \ell - \tau && (25)\\
&= n - \ell - \tau && (26)
\end{aligned}$$

In the above deduction, Equation (23) follows from the fact that $\Lambda_{sk_{id^*}}(x^*)$ is independent of pp_2, ek and s^*. Equation (24) follows from the fact that sk_S can be extracted from sp_1 and the random coins used are independent of sk_{id^*}. Equation (25) follows from the upper bound of leakage is ℓ and t^* has at most 2^τ possible values. Equation (26) follows from the ϵ_1-universal$_1$ property of the underlying IB-HPS.

For ease of analysis, we write k_β^* in Game 5 as $k_{5,\beta}^*$ and that in Game 6 as $k_{6,\beta}^*$. According to the definition of Game 5, $k_{5,0}^* \leftarrow \mathsf{ext}(\Lambda_{sk_{id^*}}(x^*), s^*)$, $k_{6,0}^* \xleftarrow{\text{R}} K$. Since ext is an average-case $(n-\ell-\tau, \epsilon_2)$-strong randomness extractor, we have $\Delta[(view', k_{5,0}^*), (view', k_{6,0}^*)] \leq \epsilon_2$. According to the definition of $k_{5,\beta}^*$ and $k_{6,\beta}^*$, we further have $\Delta[(view', k_{5,\beta}^*), (view', k_{6,\beta}^*)] \leq \epsilon_2/2$.

Observe that in Phase 2 of Game 5 and Game 6 the responses to decapsulation queries for $id \neq id^*$ are determined by sk_S and $\mathcal{A}$'s random coins, the responses to decapsulation queries for id^* are determined by id^* and $\mathcal{A}$'s random coins since the action of $\Lambda_{sk_{id^*}}$ on $x \in L_{id^*}$ is determined by id^* and the decapsulation queries $\langle id^*, c = (x, s, t)\rangle$ with $x \notin L_{id^*}$ are rejected with $\perp$. Therefore, the responses to all decapsulation queries in Phase 2 Game 5 are completely determined as a function, say h, of $(view', k_{5,\beta}^*)$, while the responses to all decapsulation queries in Phase 2 Game 6 are completely determined as the same function of $(view', k_{6,\beta}^*)$. Let $view_5 = (view', k_{5,\beta}^*, h(view', k_{5,\beta}^*))$ denote adversary's view in Game 5 and $view_6 = (view', k_{6,\beta}^*, h(view', k_{6,\beta}^*))$ denote adversary's view in Game 6. It follows that $\Delta[view_5, view_6] \leq \epsilon_2/2$. This proves the lemma. □

Putting all the above together, the theorem immediately follows. □

Discussions. One may wonder why we do not use ABO-RLF to replace OT-RLF in the above construction, which had been proved successful in the setting of KEM. Observe that toward uttermost generality, in the identity-based subset membership problem the language L could be dependent on identity, and in the security experiment for IB-KEM the attacker can choose the target identity id^* adaptively. The consequence is that the reduction is unable to determine the challenge ciphertext (more precisely, the first component $x^* \leftarrow \mathsf{SampYes}(pp, id^*)$) in advance until the attacker submits id^* in the challenge ciphertext. In line of our proof strategy, we have to rely on OT-RLF to program x^* into a lossy branch on-the-fly. However, when the language is independent of identity, as the case of realizations of [BGH07, Cor09], we do can use ABO-RLF in the place of OT-RLF to obtain more efficient and simpler construction of IB-KEM.

Acknowledgement

The first author is supported by the National Key Research and Development Plan (Grant No. 2016YFB0800403), the National Natural Science Foundation of China (Grant No. 61772522) and Youth Innovation Promotion Association CAS. The second author is supported by the National Natural Science Foundation of China (Grant No. 61502400). The third author is supported by the National Natural Science Foundation of China (Grant No. 61602473) and the National Cryptography Development Fund (Grant No. MMJJ20170116).

References

[ADN+10] Joël Alwen, Yevgeniy Dodis, Moni Naor, Gil Segev, Shabsi Walfish, and Daniel Wichs. Public-key encryption in the bounded-retrieval model. In *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 113–134. Springer, 2010.

[ADW09a] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *LNCS*, pages 36–54. Springer, 2009.

[ADW09b] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Survey: Leakage resilience and the bounded retrieval model. In *Information Theoretic Security, 4th International Conference, ICITS 2009*, pages 1–18, 2009.

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009*, volume 5444 of *LNCS*, pages 474–495. Springer, 2009.

[BFO08] Alexandra Boldyreva, Serge Fehr, and Adam O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In *Advances in Cryptology - CRYPTO 2008*, volume 5157 of *LNCS*, pages 335–359. Springer, 2008.

[BG10] Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *LNCS*, pages 1–20. Springer, 2010.

[BGH07] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *48th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2007*, pages 647–657. IEEE Computer Society, 2007.

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In *Advances in Cryptology - CRYPTO 2008*, volume 5157 of *LNCS*, pages 108–125. Springer, 2008.

[BHSV98] Mihir Bellare, Shai Halevi, Amit Sahai, and Salil P. Vadhan. Many-to-one trapdoor functions and their ralation to public-key cryptosystems. In *CRYPTO 1998*, volume 1462 of *LNCS*, pages 283–298. Springer, 1998.

[BL17] Xavier Boyen and Qinyi Li. All-but-many lossy trapdoor functions from lattices and applications. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference*, volume 10403 of *Lecture Notes in Computer Science*, pages 298–331. Springer, 2017.

[BW10] Xavier Boyen and Brent Waters. Shrinking the keys of discrete-log-type lossy trapdoor functions. In *Applied Cryptography and Network Security, 8th International Conference, ACNS 2010*, pages 35–52, 2010.

[CDRW10] Sherman S. M. Chow, Yevgeniy Dodis, Yannis Rouselakis, and Brent Waters. Practical leakage-resilient identity-based encryption from simple assumptions. In *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010*, pages 152–161. ACM, 2010.

[Cor09] Jean-Sébastien Coron. A variant of Boneh-Franklin IBE with a tight reduction in the random oracle model. *Des. Codes Cryptography*, 50(1):115–133, 2009.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, 2002.

[DHLW10] Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Efficient public-key cryptography in the presence of key leakage. In *Advances in Cryptology - ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 613–631. Springer, 2010.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[FGK+13] David Mandell Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev. More constructions of lossy and correlation-secure trapdoor functions. *J. Cryptology*, 26(1):39–74, 2013.

[Gen06] Craig Gentry. Practical identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 445–464. Springer, 2006.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC 2008*, pages 197–206. ACM, 2008.

[HK07] Dennis Hofheinz and Eike Kiltz. Secure hybrid encryption from weakened key encapsulation. In *Advances in Cryptology - CRYPTO 2007*, volume 4622 of *LNCS*, pages 553–571. Springer, 2007.

[HLAWW13] Carmit Hazay, Adriana López-Alt, Hoeteck Wee, and Daniel Wichs. Leakage-resilient cryptography from minimal assumptions. In *Advances in Cryptology - EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 160–176. Springer, 2013.

[HLOV11] Brett Hemenway, Benoît Libert, Rafail Ostrovsky, and Damien Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 70–88. Springer, 2011.

[HO12] Brett Hemenway and Rafail Ostrovsky. Extended-ddh and lossy trapdoor functions. In *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography*, volume 7293 of *LNCS*, pages 627–643. Springer, 2012.

[Hof12] Dennis Hofheinz. All-but-many lossy trapdoor functions. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 7237 of *Lecture Notes in Computer Science*, pages 209–227. Springer, 2012.

[Hof13] Dennis Hofheinz. Circular chosen-ciphertext security with compact ciphertexts. In

Advances in Cryptology - EUROCRYPT 2013, volume 7881 of *LNCS*, pages 520–536. Springer, 2013.

[KD04] Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In *Advances in Cryptology - CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442. Springer, 2004.

[KL07] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

[KMO10] Eike Kiltz, Payman Mohassel, and Adam O'Neill. Adaptive trapdoor functions and chosen-ciphertext security. In *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 673–692. Springer, 2010.

[Kom16] Ilan Komargodski. Leakage resilient one-way functions: The auxiliary-input setting. In *Theory of Cryptography - 14th International Conference, TCC 2016-B*, volume 9985 of *LNCS*, pages 139–158. Springer, 2016.

[KOS17] Eike Kiltz, Adam O'Neill, and Adam D. Smith. Instantiability of RSA-OAEP under chosen-plaintext attack. *J. Cryptology*, 30(3):889–919, 2017.

[KPSY09] Eike Kiltz, Krzysztof Pietrzak, Martijn Stam, and Moti Yung. A new randomness extraction paradigm for hybrid encryption. In *Advances in Cryptology - EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 590–609. Springer, 2009.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In *Advances in Cryptology - ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 703–720. Springer, 2009.

[LRW11] Allison B. Lewko, Yannis Rouselakis, and Brent Waters. Achieving leakage resilience through dual system encryption. In *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011*, volume 6597 of *LNCS*, pages 70–88. Springer, 2011.

[LWZ13] Shengli Liu, Jian Weng, and Yunlei Zhao. Efficient public key cryptosystem resilient to key leakage chosen ciphertext attacks. In *Topics in Cryptology - CT-RSA 2013*, volume 7779 of *LNCS*, pages 84–100. Springer, 2013.

[MY10] Petros Mol and Scott Yilek. Chosen-ciphertext security from slightly lossy trapdoor functions. In *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 296–311. Springer, 2010.

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *LNCS*, pages 18–35. Springer, 2009.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology - EUROCRYPT 1999*, volume 1592 of *LNCS*, pages 223–238. Springer, 1999.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC 2008*, pages 187–196. ACM, 2008.

[QL13] Baodong Qin and Shengli Liu. Leakage-resilient chosen-ciphertext secure public-key encryption from hash proof system and one-time lossy filter. In *Advances in Cryptology - ASIACRYPT 2013*, volume 8270 of *LNCS*, pages 381–400. Springer, 2013.

[QL14] Baodong Qin and Shengli Liu. Leakage-flexible cca-secure public-key encryption: Simple construction and free of pairing. In *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography*, volume 8383 of *LNCS*, pages 19–36. Springer, 2014.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, STOC 2005*, pages 84–93. ACM, 2005.

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009*, volume 5444 of *LNCS*, pages 419–436. Springer, 2009.

[SGL16] Shifeng Sun, Dawu Gu, and Shengli Liu. Efficient chosen ciphertext secure identity-based encryption against key leakage attacks. *Security and Communication Networks*, 9(11):1417–1434, 2016.

[Wee12] Hoeteck Wee. Dual projective hashing and its applications - lossy trapdoor functions and more. In *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 246–262. Springer, 2012.

[XLL+13] Haiyang Xue, Bao Li, Xianhui Lu, Dingding Jia, and Yamin Liu. Efficient lossy trapdoor functions based on subgroup membership assumptions. In *Cryptology and Network Security - 12th International Conference, CANS 2013*, volume 8257 of *Lecture Notes in Computer Science*, pages 235–250. Springer, 2013.

[Zha16] Mark Zhandry. The magic of elfs. In *Advances in Cryptology - CRYPTO 2016*, volume 9814 of *LNCS*, pages 479–508. Springer, 2016.

A Cryptographic Notions and Information Background

Here we recall some standard cryptographic notions and information background that will be used in this work.

A.1 One-way Functions

A family of efficiently computable functions $\mathcal{F} = \{f_i : X \to Y\}_{i \in I}$ consists of two polynomial time algorithms:

- $\mathsf{Gen}(\lambda)$: on input a security parameter λ, outputs a function index $i \in I$. Each i defines a deterministic function $f_i : X \to Y$.
- $\mathsf{Eval}(i, x)$: on input a function index i and an element $x \in X$, outputs $f_i(x)$.

Leakage-Resilient One-wayness. Let $\mathcal{A}$ be an adversary against $\mathcal{F}$ and define its advantage in the following experiment as:

$$\mathrm{Adv}_{\mathcal{A}}(\lambda) = \Pr\left[f_i(x) = y^* : \begin{array}{l} i \leftarrow \mathsf{Gen}(\lambda); \\ x^* \xleftarrow{\mathrm{R}} X, y^* \leftarrow f_i(x^*); \\ x \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{leak}}(\cdot)}(i, y^*); \end{array} \right].$$

Here $\mathcal{O}_{\mathsf{leak}}(\cdot)$ is a leakage oracle that on input $\mathsf{leak} : \{0,1\}^n \to \{0,1\}^*$ returns $\mathsf{leak}(x^*)$, and the sum of its output lengths is at most ℓ. $\mathcal{F}$ is said to be ℓ-leakage-resilient one-way if for any PPT adversary $\mathcal{A}$, its advantage defined as above is negligible in λ.

A.2 Message Authentication Codes

A MAC consists of four polynomial-time algorithms as follows.

- $\mathsf{Setup}(\lambda)$: on input a security parameter λ, output public parameter pp which includes the description of algorithm Tag, the key space K, the message space M and the tag space T.
- $\mathsf{Gen}(pp)$: on input public parameter pp, pick a random secret key $k \xleftarrow{\mathrm{R}} K$.
- $\mathsf{Tag}(k, m)$: on input secret key k and a message m, output a tag t.
- $\mathsf{Vefy}(k, m, t)$: on input secret key k and message-tag pair (m, t), output "1" indicating accept and "0" indicating reject.

Correctness. For all $pp \leftarrow \mathsf{Setup}(\lambda)$, all $k \leftarrow \mathsf{Gen}(pp)$, all $m \in M$ and correctly generated tag $t \leftarrow \mathsf{Tag}(k, m)$, we have $\mathsf{Vefy}(k, m, t) = 1$.

Leakage-Resilient Strong Unforgeability. Let $\mathcal{A}$ be an adversary against MAC and define its advantage in the following experiment:

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = \Pr\left[\begin{array}{c} \mathsf{Vefy}(k, m, t) = 1 \\ \wedge (m, t) \notin \mathcal{Q} \end{array} : \begin{array}{l} pp \leftarrow \mathsf{Setup}(\lambda); \\ k \leftarrow \mathsf{Gen}(pp); \\ (m, t) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{tag}}(\cdot), \mathcal{O}_{\mathsf{leak}}(\cdot)}(pp); \end{array} \right] \leq \mathsf{negl}(\lambda)$$

Here $\mathcal{O}_{\mathsf{tag}}(\cdot)$ is a tagging oracle that on input m returns $t \leftarrow \mathsf{Tag}(k, m)$. The set $\mathcal{Q}$ contains pairs of queries to $\mathcal{O}_{\mathsf{tag}}(\cdot)$ and their associated responses. $\mathcal{O}_{\mathsf{leak}}(\cdot)$ is a leakage oracle that on input any function $g : K \to \{0,1\}^*$ returns $g(k)$, and the sum of its output lengths is at most ℓ. MAC is said to be ℓ-leakage-resilient strongly unforgeable if no PPT adversary has non-negligible advantage in above experiment.

The strong unforgeability (sUF) for MAC can be relaxed in several ways.

- One-time: The adversary can only query the tagging oracle $\mathcal{O}_{\mathsf{tag}}(\cdot)$ once. The resulting notion is called one-time sUF.
- Selective: The adversary commits to his tagging queries selectively (a.k.a. statically) before seeing the public parameter. The resulting notion is called selective sUF.

We also consider a combination of the above restrictions, yielding selective one-time sUF.

Remark A.1. The above security definition does not incorporate a verification oracle $\mathcal{O}_{\mathsf{vefy}}(\cdot,\cdot)$ that on input (m,t) returns $\mathsf{Vefy}(k,m,t)$. It turns out the definitions with and without $\mathcal{O}_{\mathsf{vefy}}(\cdot,\cdot)$ are actually equivalent for strong unforgeability [KL07], and the equivalence even holds in the setting of leakage [HLAWW13]. As such, in this work we choose the definition without $\mathcal{O}_{\mathsf{vefy}}(\cdot,\cdot)$ for simplicity of presentation.

A.3 Key Encapsulation Mechanism

A key encapsulation mechanism (KEM) consists of four polynomial time algorithms as follows.

- $\mathsf{Setup}(\lambda)$: on input a security parameter λ, output global public parameter pp which includes the descriptions of public key space PK, secret key space SK, ciphertext space C, key space K.
- $\mathsf{KeyGen}(\lambda)$: on input a security parameter λ, output a public key pk and a secret key sk.
- $\mathsf{Encaps}(pk)$: on input a public key pk, output a ciphertext $c \in C$ and a DEM key $k \in K$.
- $\mathsf{Decaps}(sk,c)$: on input a secret key sk and a ciphertext $c \in C$, output a DEM key $k \in K$ or an distinguished symbol $\perp$ indicating c is invalid.

Correctness. For $pp \leftarrow \mathsf{Setup}(\lambda)$ and $(pk,sk) \leftarrow \mathsf{KeyGen}(pp)$ and $(c,k) \leftarrow \mathsf{Encaps}(pk)$, then $\mathsf{Decaps}(sk,c) = k$ with all but negligible probability over all the randomness in the experiment.

Leakage-Resilient CCA. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary against KEM and define its advantage in the following experiment:

$$\mathrm{Adv}_{\mathcal{A}}(\lambda) = \Pr\left[\beta = \beta' : \begin{array}{l} pp \leftarrow \mathsf{Setup}(\lambda); \\ (pk,sk) \leftarrow \mathsf{KeyGen}(pp); \\ state \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{leak}}(\cdot)}(pk); \\ (c^*,k_0^*) \leftarrow \mathsf{Encaps}(pk), k_1^* \xleftarrow{\mathrm{R}} K; \\ \beta \xleftarrow{\mathrm{R}} \{0,1\}; \\ \beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{decaps}}(\cdot)}(state, c^*, k_\beta^*); \end{array}\right] - \frac{1}{2}.$$

Here $\mathcal{O}_{\mathsf{leak}}(\cdot)$ is a leakage oracle that on input $f : SK \to \{0,1\}^*$ returns $f(sk)$, and the sum of its output lengths is at most ℓ. $\mathcal{O}_{\mathsf{decaps}}(\cdot)$ is a decapsulation oracle that on input $c \in C$ where $c \neq c^*$ returns $\mathsf{Decaps}(sk,c)$. KEM is said to be ℓ-leakage-resilient against chosen-ciphertext attack (LR-CCA) if for any PPT adversary $\mathcal{A}$, its advantage defined as above is negligible in λ.

A.4 Randomness Extraction

The statistical distance between two random variables X and Y over a finite domain Ω is defined as $\Delta(X,Y) = \frac{1}{2}\left|\Pr[X=\omega] \Pr[Y=\omega]\right|$.

Let $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ denote two ensembles of random variables indexed by λ. We say that X and Y are statistically indistinguishable, written $X \approx_s Y$, if $\Delta(X_\lambda, Y_\lambda) = \mathsf{negl}(\lambda)$. We say that X and Y are computationally indistinguishable, written $X \approx_c Y$, if the advantage of any PPT algorithm in distinguishing X_λ and Y_λ is $\mathsf{negl}(\lambda)$.

The min-entropy of a random variable X over a domain Ω is the negative (base-2) logarithm of the *unpredictability* of X:

$$\mathsf{H}_\infty(X) = -\log\left(\max_{\omega\in\Omega}\Pr[X=\omega]\right).$$

In many natural settings, the variable X is correlated with another variable Y whose value is known to an adversary. In such scenarios, it is most convenient to use the notion of *average min-entropy* as defined by Dodis et al. [DORS08], which captures the *average unpredictability* of X conditioned Y:

$$\tilde{\mathsf{H}}_\infty(X|Y) = -\log\left(\mathrm{E}_{y\leftarrow Y}\left[2^{\mathsf{H}_\infty(X|Y=y)}\right]\right) = -\log\left(\mathrm{E}_{y\leftarrow Y}\left[\max_{\omega\in\Omega}\Pr[X=\omega|Y=y]\right]\right)$$

The average min-entropy corresponds to the optimal probability of guessing X, given knowledge of Y. The following bound of average min-entropy was proved in [DORS08].

Lemma A.1 ([DORS08])**.** *Let X, Y, Z be random variables. If Z has 2^r possible values, $\tilde{\mathsf{H}}_\infty(X|(Y,Z)) \geq \tilde{\mathsf{H}}_\infty(X|Y) - r$. In particular, $\tilde{\mathsf{H}}_\infty(X|Y) \geq \mathsf{H}_\infty(X) - r$.*

In cryptographic applications, we usually need to derive nearly uniform bits from a weakly random source X that has some average min-entropy. This is accomplished via an appropriate type of *randomness extractor*.

Definition A.1 ([DORS08])**.** A function $\mathsf{ext} : \Omega\times\{0,1\}^d \to \{0,1\}^k$ is an average-case (m,k,ϵ)-strong extractor if for all pairs of random variables (X,Y) such that $X\in\Omega$ and $\tilde{\mathsf{H}}_\infty(X|Y)\geq m$, it holds that:

$$\Delta((\mathsf{ext}(X,S),S,Y),(K,S,Y)) \leq \epsilon,$$

where S is uniform over $\{0,1\}^d$ and K is uniform over $\{0,1\}^k$.

Dodis et al. [DORS08] proved that any strong extractor is in fact an average-case strong extractor for appropriate setting of the parameters. As a specific example, they proved that any family of universal hash functions is an average-case strong extractor.

Lemma A.2 ([DORS08])**.** *Let X, Y be random variables such that $\tilde{\mathsf{H}}_\infty(X|Y)\geq m$. Let $\mathcal{H}$ be a family of universal hash functions from X to K. Then for $h \xleftarrow{\mathrm{R}} \mathcal{H}$, it holds that:*

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

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

B Instantiations of Algebraic Subset Membership Assumptions

We first present instantiations of ASM assumptions from the DDH and d-linear assumptions. Let $\mathsf{GroupGen}$ be a PPT algorithm that takes as input a security parameter λ, and outputs a triplet $(\mathbb{G},q,g)$ where $\mathbb{G}$ is a group of order q that is generated by $g\in\mathbb{G}$, and q is a λ-bit prime number.

Instantiation under the DDH assumption. The decisional Diffie-Hellman (DDH) assumption is that (g_1,g_2,g_1^r,g_2^r) and $(g_1,g_2,g_1^{r_1},g_2^{r_2})$ are computationally indistinguishable, where $(\mathbb{G},q,g)\leftarrow\mathsf{GroupGen}(\lambda)$, and the elements g_1,g_2 and $r,r_1,r_2\in\mathbb{Z}_q$ are chosen independently and uniformly at random.

The DDH assumption leads to the immediate instantiation of the ASM assumption by setting $X=\mathbb{G}\times\mathbb{G}$, $L=\{(g_1^r,g_2^r) : r\in\mathbb{Z}_q\}$. The order of H is a prime q and thus H is cyclic.

Instantiation under the d-Linear assumption. The d-Linear assumption for every positive integer $d \geq 1$ is that $(g_1, \cdots, g_d, g_{d+1}, g_1^{r_1}, \ldots, g_d^{r_d}, g_{d+1}^{\sum_{i=1}^{d} r_i})$ and $(g_1, \cdots, g_d, g_{d+1}, g_1^{r_1}, \ldots, g_d^{r_d}, g_{d+1}^{r_i})$ are computationally indistinguishable, where $(\mathbb{G}, q, g) \leftarrow \mathsf{GroupGen}(\lambda)$, and $g_1, \ldots, g_{d+1} \in \mathbb{G}$ and $r_1, \ldots, r_{d+1} \in \mathbb{Z}_q$ are chosen independently and uniformly at random. Note that DDH is the 1-Linear assumption.

This d-Linear assumption leads to the immediate instantiation of the SGMP by setting $X = \mathbb{G}^{d+1}$, $L = \{(g_1^{r_1}, \ldots, g_d^{r_d}, g_{d+1}^{\sum_{i=1}^{d} r_i}) : r_1, \ldots, r_d \in \mathbb{Z}_q\}$. The order of H is a prime q and thus H is cyclic.

We then present instantiations from the QR and DCR assumptions. Let $\mathsf{GenModulus}$ be a PPT algorithm that takes as input a security parameter λ, and outputs a triplet (N, p, q) where N is a Blum integer, p and q are two random λ-bit prime number such that $p, q \equiv 3 \bmod 4$.

Instantiation under the QR assumption. Let $\mathbb{J}_N$ be the set of elements in $\mathbb{Z}_N^*$ with Jacobi symbol 1, $\mathbb{QR}_N$ be the set of quadratic residues (squares) modulo N. The quadratic residuosity (QR) assumption is that for $(N, p, q) \leftarrow \mathsf{GroupGen}(\lambda)$, the uniform distribution over $\mathbb{J}_N$ and $\mathbb{QR}_N$ are computationally indistinguishable.

The QR assumption leads to the immediate instantiation of the SGMP by setting $X = \mathbb{J}_N$, $L = \mathbb{QR}_N$. The order of H is a prime 2 and thus H is cyclic.

Instantiation under the DCR assumption. The decisional composite residuosity (DCR) assumption is that for a properly generated RSA number N, the uniform distribution over $\mathbb{Z}_{N^2}^*$ and its subgroup of N^{th}-residues $\{x^N : x \in \mathbb{Z}_{N^2}^*\}$ are computationally indistinguishable.

The DCR assumption leads to the immediate instantiation of the SGMP by setting $X = \mathbb{Z}_{N^2}^*$, $L = \{x^N : x \in \mathbb{Z}_{N^2}^*\}$. The order of X is $N\phi(N)$, the order of L is $\phi(N)$, and H is a cyclic group with order N.