Leakage-Resilient Chosen-Ciphertext Secure Public-Key Encryption from Hash Proof System and One-Time Lossy Filter [*]

Baodong Qin [†] Shengli Liu [‡]

October 9, 2013

Abstract

We present a new generic construction of a public-key encryption (PKE) scheme secure against leakage-resilient chosen-ciphertext attacks (LR-CCA), from any Hash Proof System (HPS) and any one-time lossy filter (OT-LF). Efficient constructions of HPSs and OT-LFs from the DDH and DCR assumptions suggest that our construction is a practical approach to LR-CCA security. Most of practical PKEs with LR-CCA security, like variants of Cramer-Shoup scheme, rooted from Hash Proof Systems, but with leakage rates at most $1/4 - o(1)$ (defined as the ratio of leakage amount to secret-key size). The instantiations of our construction from the DDH and DCR assumptions result in LR-CCA secure PKEs with leakage rate of $1/2 - o(1)$. On the other hand, our construction also creates a new approach for constructing IND-CCA secure (leakage-free) PKE schemes, which may be of independent interest.

Keywords: Public-key encryption, leakage-resilience, chosen-ciphertext security, hash proof system

1 Introduction

Research on leakage-resilient cryptography is motivated by those side-channel attacks [17], in which a significant fraction of the secret key SK is leaked to the adversary. Cryptosystems proved secure in the traditional model may suffer from these key-leakage attacks, as shown in [17]. This fact leads to design and security proof of a variety of leakage-resilient cryptosystems, including stream ciphers [14, 30], block ciphers [12], digital signatures [20, 15], public key encryption [27, 1, 2, 3, 4], identity-based encryption [24, 7, 16], etc.

Leakage Oracle, Bounded-Leakage Model and Leakage Rate. Side-channel attacks characterized by key leakage can be formalized in a general framework [1] with a leakage oracle $\mathcal{O}_{SK}^{\lambda,\kappa}(\cdot)$: the adversary queries arbitrary efficiently computable functions $f_i : \{0,1\}^* \to \{0,1\}^{\lambda_i}$ of the secret key SK repeatedly and adaptively, and the leakage oracle responds with $f_i(SK)$. The *bounded-leakage* model limits the total amount of information about SK leaked by the oracle to a bound λ during the life time of the cryptosystem. This model is simple and powerful, but a thorough understanding of this model is essential to those more complicated models [4]. If a cryptosystem is secure against the above key-leakage attacks, we call it λ-leakage-resilient (λ-LR, for short). The *leakage rate* is defined as the ratio of λ to the secret key size, i.e., $\lambda/|SK|$.

Leakage-Resilient CCA Security and Hash Proof System. In the key-leakage scenario of public key encryption (PKE), leakage-resilient security against chosen-plaintext attacks (LR-CPA) is characterized by the indistinguishability between the encryptions of two plaintexts (of equal length) chosen by any Probabilistic Polynomial-Time (PPT) adversary, who is given access to a key-leakage oracle. If the

[*]This is the full version of Asiacrypt 2013 paper.

[†]Shanghai Jiao Tong University and Southwest University of Science and Technology. qinbaodong@sjtu.edu.cn

[‡]Shanghai Jiao Tong University. slliu@sjtu.edu.cn

adversary is equipped with a decryption oracle as well, with restriction that the challenge ciphertext is refused by the decryption oracle and the leakage oracle stops working after the generation of the challenge ciphertext, the notion becomes leakage-resilient security against chosen-ciphertext attacks (LR-CCA). Naor-Yung paradigm applies to LR-CCA security [27]. It achieves leakage rate of $1 - o(1)$, but the simulation-sound Non-Interactive Zero-Knowledge (ss-NIZK) proof is far from practical. It was later improved by Dodis et al. [11] with true-simulation extractable NIZK (tSE-NIZK), but the construction is still not practical. Recently, Galindo et al. [16] constructed an identity-based encryption (IBE) scheme with master key-dependent chosen-plaintext (mKDM-sID-CPA) security based on the decisional linear assumption over bilinear groups. They suggested that their mKDM-sID-CPA secure IBE scheme is also master key leakage resilient with rate $1 - o(1)$, hence can be transformed into a LR-CCA secure PKE scheme with leakage rate $1 - o(1)$ by applying the CHK transform [6]. However, their claim that the mKDM-sID-CPA secure IBE scheme is also master key leakage resilient was not supported by any rigorous proof.

Hash Proof Systems (HPSs), due to Cramer and Shoup [9], have long been served as the most practical approach to PKEs with IND-CCA security. They are also intrinsically LR-CPA secure, and a HPS based on the DDH assumption (and its d-Linear variant) was proved to be LR-CPA secure with leakage rate of $1 - o(1)$ [27]. As to LR-CCA security, however, the HPS approach to IND-CCA security is inherently limited to leakage rate below $1/2$, as pointed out by Dodis et al. [11]. Recall that to achieve IND-CCA security, Cramer and Shoup [9] proposed to use two independent HPSs, one is a smooth HPS to mask and hide the plaintext, and the other is a universal$_2$ HPS used to verify whether the ciphertext is well-formed. Hence two independent secret keys are involved in the construction, and either one, if totally leaked, will kill the LR-CCA security. That is why the leakage rate must be less than $1/2$.

Prior constructions of PKE with LR-CCA security from HPSs enjoy great efficiency, but suffer from low leakage rate. The variants [27, 26] of Cramer-Shoup DDH-based scheme [8] achieve leakage rate of $1/6 - o(1)$, which was later improved to $1/4 - o(1)$ [25]. To the best of our knowledge, no constructions from HPSs are known to be LR-CCA secure with leakage rate of $1/2 - o(1)$. The question is: can we find a new way to construct LR-CCA secure PKEs which are not only as practical as HPS but also with reasonable high leakage rates (like $1/2 - o(1)$)?

Our Contributions. We propose a new generic construction of PKE with LR-CCA security from a Hash Proof System (HPS) and a one-time lossy filter (OT-LF). The new primitive, one-time lossy filter (OT-LF), is a weak version of lossy algebraic filter [19], and we show how to construct OT-LFs from the DDH and DCR assumptions. In the generic construction of LR-CCA secure PKE, the HPS is used to generate an encapsulated key K, which is not only used to mask the plaintext, but also used in the OT-LF to verify the well-formedness of ciphertexts. OT-LF helps to obtain a higher leakage rate, compared to the constructions solely from HPSs.

- We give instantiations of PKEs with LR-CCA security under the DDH (DCR) assumption, by combining an efficient construction of DDH (DCR)-based OT-LF and DDH (DCR)-based HPS. The leakage rate is as high as $1/2 - o(1)$.
- In case of no leakage on secret key at all, the leakage-free version of our construction opens another practical approach to IND-CCA security, as compared to the HPS-based construction by Cramer and Shoup.

Overview of Our Techniques. Different from the HPS-based approach to CCA-security, in which a universal$_2$ hash proof system is employed to reject ill-formed ciphertexts, we use a one-time lossy filter (OT-LF) to do the job. OT-LF is a simplified version of lossy algebraic filter, which was introduced by Hofheinz [19] recently to realize key-dependent chosen-ciphertext security [5]. The concept of OT-LF is similar to (chameleon) all-but-one lossy trapdoor function [31, 23]. But it does not require efficient inversion. Roughly, a OT-LF is a family of functions indexed by a public key Fpk and a tag $t = (t_a, t_c)$. A function $\mathsf{LF}_{Fpk,t}(\cdot)$ from that family maps an input X to a unique output. For a fixed public key, the set of tags contains two computationally indistinguishable disjoint subsets, namely the subset of injective tags and the subset of lossy ones. If tag $t = (t_a, t_c)$ is injective, then so is the corresponding function $\mathsf{LF}_{Fpk,t}(\cdot)$. If the tag is lossy, the output of the function reveals only a constant amount of information about its input X. For any t_a, there exists a lossy tag (t_a, t_c) such that t_c can be efficiently computed through a trapdoor Ftd. Without this trapdoor, however, it is hard to generate a new lossy tag even with the knowledge of one lossy tag. Trapdoor Ftd and lossy tag are only used for the security proof.

Roughly speaking, a hash proof system HPS is a key-encapsulation mechanism. Given public key pk, an element $C \in \mathcal{V}$ and its witness w, the encapsulated key is given by $K = \mathsf{HPS.Pub}(pk, C, w)$. With secret key sk, decapsulation algorithm $\mathsf{HPS.Priv}(sk, C)$ recovers K from $C \in \mathcal{V}$. If $C \in \mathcal{C} \setminus \mathcal{V}$, the output of $\mathsf{HPS.Priv}(sk, C)$ has a high min-entropy even conditioned on pk and C. The hardness of subset membership problem requires that elements in $\mathcal{V}$ are indistinguishable from those in $\mathcal{C} \setminus \mathcal{V}$.

In our construction, the secret key is just sk from the HPS, and the HPS and OT-LF are integrated into a ciphertext CT,

$$CT = (C, \quad s, \quad \Psi = \mathsf{Ext}(K, s) \oplus M, \quad \Pi = \mathsf{LF}_{Fpk,t}(K), \quad t_c),$$

via $K = \mathsf{HPS.Pub}(pk, C, w) = \mathsf{HPS.Priv}(sk, C)$ (it holds for all $C \in \mathcal{V}$).

The encapsulated key K functions in two ways. (1) It serves as an input, together with a random string s, to extractor $\mathsf{Ext}(K, s)$ to mask and hide the plaintext M to deal with key leakage. (2) It serves as the input of $\mathsf{LF}_{Fpk,t}(\cdot)$ to check the well-formedness of the ciphertext. Tag $t = (t_a, t_c)$ is determined by $t_a = (C, s, \Psi)$ and a random t_c. $\mathsf{LF}_{Fpk,t}(K)$ can also be considered as an authentication code, which is used to authenticate the tag $t = ((C, s, \Psi), t_c)$ with the authentication key K.

In the security proof, some changes are made to the generation of the challenge ciphertext $CT^* = (C^*, s^*, \Psi^*, \Pi^*, t_c^*)$: C^* is sampled from $\mathcal{C} \setminus \mathcal{V}$ and the tag t^* is made lossy by computing a proper t_c with trapdoor Ftd. A PPT adversary cannot tell the changes due to the hardness of subset membership problem and the indistinguishability of lossy tags and injective ones. Conditioned on CT^*, the encapsulated key $K^* = \mathsf{HPS.Priv}(sk, C^*)$ still maintains a high min-entropy since $\Pi^* = \mathsf{LF}_{Fpk,t^*}(K^*)$ works in lossy mode and only little information is released. When a PPT adversary chooses an invalid ciphertext CT in the sense that $C \in \mathcal{C} \setminus \mathcal{V}$ for decryption query, the corresponding tag t is injective with overwhelming probability. Then $\mathsf{LF}_{Fpk,t}(\cdot)$ is injective and Π preserves the high min-entropy of $K = \mathsf{HPS.Priv}(sk, C)$. Hence invalid ciphertexts will be rejected by the decryption oracle with overwhelming probability. On the other hand, the information of pk has already determined $K = \mathsf{HPS.Priv}(sk, C)$ for all $C \in \mathcal{V}$. Thus the decryption oracle does not help the adversary to gain any more information about K^*. Then an extractor can be applied to K^* to totally mask the information of challenge plaintext, and a large min-entropy of K^* conditioned on pk and Π^* implies a high tolerance of key leakage.

Thanks to efficient constructions for HPS and OT-LF under the DDH and DCR assumptions, the instantiations are practically efficient. More precisely, $|K| \approx L/2$, where L is the length of the secret key of HPS. Due to the lossiness of the OT-LF and the property of the HPS, the min-entropy conditioned on the public key and challenge ciphertext, approaches $(1/2 - o(1))L$. Hence the leakage rate approaches $1/2$.

2 Preliminaries

Notation. Let $[n]$ denote the set $\{1, \ldots, n\}$. Let $\kappa \in \mathbb{N}$ denote the security parameter and 1^κ denote the string of κ ones. If s is a string, then $|s|$ denotes its length, while if S is a set then $|S|$ denotes its size and $s \leftarrow S$ denotes the operation of picking an element s uniformly at random from S. We denote $y \leftarrow A(x)$ the operation of running A with input x, and assigning y as the result. We write $\log s$ for logarithms over the reals with base 2.

Randomness Extractor. Let $\mathrm{SD}(X, Y)$ denote the *statistical distance* of random variables X and Y over domain Ω. Namely, $\mathrm{SD}(X, Y) = \frac{1}{2} \sum_{\omega \in \Omega} |\Pr[X = \omega] - \Pr[Y = \omega]|$. The *min-entropy* of X is $\mathrm{H}_\infty(X) = -\log(\max_{\omega \in \Omega} \Pr[X = \omega])$. Dodis et al. [13] formalized the notion of *average min-entropy* of X conditioned on Y which is defined as $\widetilde{\mathrm{H}}_\infty(X|Y) = -\log(E_{y \leftarrow Y}[2^{-\mathrm{H}_\infty(X|Y=y)}])$. They proved the following property of average min-entropy.

Lemma 1. *[13] Let X, Y and Z be random variables. If Y has at most 2^r possible values, then* $\widetilde{\mathrm{H}}_\infty(X|(Y,Z)) \geq \widetilde{\mathrm{H}}_\infty(X|Z) - r$.

Definition 1 (Randomness Extractor). *An efficient function $\mathsf{Ext} : \mathcal{X} \times \mathcal{S} \to \mathcal{Y}$ is an average-case (ν, ϵ)-strong extractor if for all pairs of random variables (X, Z) such that $X \in \mathcal{X}$ and $\widetilde{\mathrm{H}}_\infty(X|Z) \geq \nu$, we have*

$$\mathrm{SD}((Z, s, \mathsf{Ext}(X, s)), (Z, s, U_{\mathcal{Y}})) \leq \epsilon,$$

where s is uniform over $\mathcal{S}$ and $U_{\mathcal{Y}}$ is uniform over $\mathcal{Y}$.

A family of universal hash functions $\mathcal{H} = \{H_s : \mathcal{X} \to \mathcal{Y}\}_{s \in \mathcal{S}}$ can be used as an average-case $(\widetilde{\mathrm{H}}_\infty(X|Z), \epsilon)$-strong extractors whenever $\widetilde{\mathrm{H}}_\infty(X|Z) \geq \log|\mathcal{Y}| + 2\log(1/\epsilon)$, according to the general Leftover Hash Lemma [13].

2.1 Leakage-Resilient Public-Key Encryption

A *Public-Key Encryption* (PKE) scheme with plaintext space $\mathcal{M}$ is given by three PPT algorithms $(\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$. The key generation algorithm $\mathsf{PKE.Gen}$ takes as input 1^κ, and outputs a pair of public/secret keys (PK, SK). The encryption algorithm $\mathsf{PKE.Enc}$ takes as input a public key PK and a plaintext $M \in \mathcal{M}$, and returns a ciphertext $CT = \mathsf{PKE.Enc}(PK, M)$. The decryption algorithm $\mathsf{PKE.Dec}$ takes as input a secret key SK and a ciphertext CT, and returns a plaintext $M \in \mathcal{M} \cup \{\bot\}$. For consistency, we require that $\mathsf{PKE.Dec}(SK, \mathsf{PKE.Enc}(PK, M)) = M$ holds for all $(PK, SK) \leftarrow \mathsf{PKE.Gen}(1^\kappa)$ and all plaintexts $M \in \mathcal{M}$.

Following [27, 28], we define leakage-resilient chosen-ciphertext security (LR-CCA) for PKE.

Definition 2 (Leakage-Resilient CCA security of PKE). *A public-key encryption scheme* PKE=($\mathsf{PKE.Gen}$, $\mathsf{PKE.Enc}$, $\mathsf{PKE.Dec}$) *is λ-leakage-resilient chosen-ciphertext secure (λ-LR-CCA-secure), if for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the function* $\mathsf{Adv}^{\text{lr-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa) := \left|\Pr[\mathsf{Exp}^{\text{lr-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa) = 1] - \frac{1}{2}\right|$ *is negligible in κ. Below defines* $\mathsf{Exp}^{\text{lr-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)$.

1. $(PK, SK) \leftarrow \mathsf{PKE.Gen}(1^\kappa)$, $b \leftarrow \{0, 1\}$.
2. $(M_0, M_1, state) \leftarrow \mathcal{A}_1^{\mathcal{O}^{\lambda,\kappa}_{sk}(\cdot), \mathsf{PKE.Dec}(SK,\cdot)}(pk)$, *s.t.* $|M_0| = |M_1|$.
3. $CT^* \leftarrow \mathsf{PKE.Enc}(PK, M_b)$. $b' \leftarrow \mathcal{A}_2^{\mathsf{PKE.Dec}_{\neq CT^*}(SK,\cdot)}(state, CT^*)$.
5. *If $b = b'$ return 1 else return 0.*

In the case of $\lambda = 0$, Definition 2 is just the standard CCA security [32].

2.2 Hash Proof System

We recall the notion of hash proof systems introduced by Cramer and Shoup [9]. For simplicity, hash proof systems are described as key encapsulation mechanisms (KEMs), as did in [21].

Projective Hashing. Let $\mathcal{SK}$, $\mathcal{PK}$ and $\mathcal{K}$ be sets of public keys, secret keys and encapsulated keys. Let $\mathcal{C}$ be the set of all ciphertexts of KEM and $\mathcal{V} \subset \mathcal{C}$ be the set of all *valid* ones. We assume that there are efficient algorithms for sampling $sk \leftarrow \mathcal{SK}$, $C \leftarrow \mathcal{V}$ together with a witness w, and $C \leftarrow \mathcal{C} \setminus \mathcal{V}$.

Let $\Lambda_{sk} : \mathcal{C} \to \mathcal{K}$ be a hash function indexed with $sk \in \mathcal{SK}$ that maps ciphertexts to symmetric keys. The hash function Λ_{sk} is *projective* if there exists a projection $\mu : \mathcal{SK} \to \mathcal{PK}$ such that $\mu(sk) \in \mathcal{PK}$ defines the action of Λ_{sk} over the subset $\mathcal{V}$ of valid ciphertexts.

Definition 3 (universal[9]). *A projective hash function Λ_{sk} is ϵ-universal, if for all pk, $C \in \mathcal{C} \setminus \mathcal{V}$, and all $K \in \mathcal{K}$, it holds that* $\Pr[\Lambda_{sk}(C) = K \mid (pk, C)] \leq \epsilon$, *where the probability is over all possible $sk \leftarrow \mathcal{SK}$ with $pk = \mu(sk)$.*

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

Lemma 2. *Assume that $\Lambda_{sk} : \mathcal{C} \to \mathcal{K}$ is an ϵ-universal projective hash function. Then, for all pk and $C \in \mathcal{C} \setminus \mathcal{V}$, it holds that $\mathrm{H}_\infty(\Lambda_{sk}(C)|(pk, C)) \geq \log 1/\epsilon$, where $sk \leftarrow \mathcal{SK}$ with $pk = \mu(sk)$.*

Hash Proof System. A hash proof system HPS consists of three PPT algorithms ($\mathsf{HPS.Gen}$, $\mathsf{HPS.Pub}$, $\mathsf{HPS.Priv}$). The parameter generation algorithm $\mathsf{HPS.Gen}(1^\kappa)$ generates parameterized instances of the form params=(group, $\mathcal{K}$, $\mathcal{C}$, $\mathcal{V}$, $\mathcal{SK}$, $\mathcal{PK}$, $\Lambda_{(\cdot)} : \mathcal{C} \to \mathcal{K}$, $\mu : \mathcal{SK} \to \mathcal{PK}$), where group may contain additional structural parameters. The public evaluation algorithm $\mathsf{HPS.Pub}(pk, C, w)$ takes as input a projective public key $pk = \mu(sk)$, a valid ciphertext $C \in \mathcal{V}$ and a witness w of the fact that $C \in \mathcal{V}$, and computes the encapsulated key $K = \Lambda_{sk}(C)$. The private evaluation algorithm $\mathsf{HPS.Priv}(sk, C)$ takes a secret key sk and a ciphertext $C \in \mathcal{V}$ as input, and returns the encapsulated key $K = \Lambda_{sk}(C)$ without knowing a witness. We assume that μ and $\Lambda_{(\cdot)}$ are efficiently computable.

Subset Membership Problem. The *subset membership problem* associated with a HPS suggests that a random valid ciphertext $C_0 \leftarrow \mathcal{V}$ and a random invalid ciphertext $C_1 \leftarrow \mathcal{C} \setminus \mathcal{V}$ are computationally indistinguishable. This is formally captured by a negligible advantage function $\mathsf{Adv}^{\mathrm{smp}}_{\mathsf{HPS},\mathcal{A}}(\kappa)$ for all PPT adversary $\mathcal{A}$, where

$$\mathsf{Adv}^{\mathrm{smp}}_{\mathsf{HPS},\mathcal{A}}(\kappa) = |\Pr[\mathcal{A}(\mathcal{C}, \mathcal{V}, C_0) = 1 \mid C_0 \leftarrow \mathcal{V}] - \Pr[\mathcal{A}(\mathcal{C}, \mathcal{V}, C_1) = 1 \mid C_1 \leftarrow \mathcal{C} \setminus \mathcal{V}]| .$$

Definition 4. *A hash proof system $\mathsf{HPS} = (\mathsf{HPS.Gen}, \mathsf{HPS.Pub}, \mathsf{HPS.Priv})$ is ϵ-universal if: (i) for all sufficiently large $\kappa \in \mathbb{N}$ and for all possible outcomes of $\mathsf{HPS.Gen}(1^\kappa)$, the underlying projective hash function is $\epsilon(\kappa)$-universal for negligible $\epsilon(\kappa)$; (ii) the underlying subset membership problem is hard. Furthermore, a hash proof system is called perfectly universal if $\epsilon(\kappa) = 1/|\mathcal{K}|$.*

2.3 One-time Lossy Filter

One-time Lossy Filter (OT-LF) is a simplified version of lossy algebraic filters recently introduced by Hofheinz [19]. A $(\mathsf{Dom}, \ell_{\mathsf{LF}})$-OT-LF is a family of functions indexed by a public key Fpk and a tag t. A function $\mathsf{LF}_{Fpk,t}$ from the family maps an input $X \in \mathsf{Dom}$ to an output $\mathsf{LF}_{Fpk,t}(X)$. Given public key Fpk, the set of tags $\mathcal{T}$ contains two computationally indistinguishable disjoint subsets, namely the subset of injective tags $\mathcal{T}_{inj}$ and the subset of lossy ones $\mathcal{T}_{loss}$. If t is an injective tag, the function $\mathsf{LF}_{Fpk,t}$ is injective and has image size of $|\mathsf{Dom}|$. If t is lossy, the output of the function has image size at most $2^{\ell_{\mathsf{LF}}}$. Thus, a lossy tag ensures that $\mathsf{LF}_{Fpk,t}(X)$ reveals at most ℓ_{LF} bits of information about its input X. This is a crucial property of an LF.

Definition 5 (OT-LF)**.** *A $(\mathsf{Dom}, \ell_{\mathsf{LF}})$-one-time lossy filter LF consists of three PPT algorithms ($\mathsf{LF.Gen}$, $\mathsf{LF.Eval}$, $\mathsf{LF.LTag}$):*

Key Generation. $\mathsf{LF.Gen}(1^\kappa)$ *outputs a key pair (Fpk, Ftd). The public key Fpk defines a tag space $\mathcal{T} = \{0,1\}^* \times \mathcal{T}_c$ that contains two disjoint subsets, the subset of lossy tags $\mathcal{T}_{loss} \subseteq \mathcal{T}$ and that of injective tags $\mathcal{T}_{inj} \subseteq \mathcal{T}$. A tag $t = (t_a, t_c) \in \mathcal{T}$ consists of an auxiliary tag $t_a \in \{0,1\}^*$ and a core tag $t_c \in \mathcal{T}_c$. Ftd is a trapdoor that allows to efficiently sample a lossy tag.*

Evaluation. $\mathsf{LF.Eval}(Fpk, t, X)$*, for a public key Fpk, a tag t and $X \in \mathsf{Dom}$, computes $\mathsf{LF}_{Fpk,t}(X)$.*

Lossy Tag Generation. $\mathsf{LF.LTag}(Ftd, t_a)$*, for an auxiliary tag t_a and the trapdoor Ftd, computes a core tag t_c such that $t = (t_a, t_c)$ is lossy.*
We require that an OT-LF LF has the following properties:

Lossiness. *If t is injective, so is the function $\mathsf{LF}_{Fpk,t}(\cdot)$. If t is lossy, then $\mathsf{LF}_{Fpk,t}(X)$ has image size of at most $2^{\ell_{\mathsf{LF}}}$. (In application, we are interested in OT-LFs that have a constant parameter ℓ_{LF} even for larger domain.)*

Indistinguishability. *For any PPT adversary $\mathcal{A}$, it is hard to distinguish a lossy tag from a random tag, i.e., the following advantage is negligible in κ.*

$$\mathsf{Adv}^{\mathrm{ind}}_{\mathsf{LF},\mathcal{A}}(\kappa) := |\Pr[\mathcal{A}(Fpk, (t_a, t_c^{(0)})) = 1] - \Pr[\mathcal{A}(Fpk, (t_a, t_c^{(1)})) = 1|$$

where $(Fpk, Ftd) \leftarrow \mathsf{LF.Gen}(1^\kappa)$, $t_a \leftarrow \mathcal{A}(Fpk)$, $t_c^{(0)} \leftarrow \mathsf{LF.LTag}(Ftd, t_a)$ *and* $t_c^{(1)} \leftarrow \mathcal{T}_c$.

Evasiveness. *For any PPT adversary $\mathcal{A}$, it is hard to generate a non-injective tag*[1] *even given a lossy tag, i.e., the following advantage is negligible in κ.*

$$\mathsf{Adv}^{\mathrm{eva}}_{\mathsf{LF},\mathcal{A}}(\kappa) := \Pr\left[\begin{array}{c}(t'_a, t'_c) \neq (t_a, t_c) \wedge \\ (t'_a, t'_c) \in \mathcal{T} \setminus \mathcal{T}_{inj}\end{array} : \begin{array}{l}(Fpk, Ftd) \leftarrow \mathsf{LF.Gen}(1^\kappa); \\ t_a \leftarrow \mathcal{A}(Fpk); t_c \leftarrow \mathsf{LF.LTag}(Ftd, t_a); \\ (t'_a, t'_c) \leftarrow \mathcal{A}(Fpk, (t_a, t_c))\end{array}\right]$$

Remark 1. The definition of one-time lossy filter is different from that of lossy algebraic filter [19] in two ways. First, the one-time property in our definition allows the adversary to query lossy tag generation oracle only once in both indistinguishability and evasiveness games. While in [19], the adversary is allowed to query the oracle polynomial times. Secondly, unlike lossy algebraic filter, one-time lossy filter does not require any algebraic properties.

2.4 Chameleon Hashing

A chameleon hashing function [22] is essentially a hashing function associated with a pair of evaluation key and trapdoor. Its collision-resistant property holds when only the evaluation key of the function is known, but is broken with the trapdoor. We recall the formal definition of chameleon hashing from [18].

Definition 6 (Chameleon Hashing)**.** *A chameleon hashing function* CH *consists of three PPT algorithms* $(\mathsf{CH.Gen}, \mathsf{CH.Eval}, \mathsf{CH.Equiv})$*:*

Key Generation. $\mathsf{CH.Gen}(1^\kappa)$ *outputs an evaluation key ek_{CH} and a trapdoor td_{CH}.*

Evaluation. $\mathsf{CH.Eval}(ek_{\mathrm{CH}}, x; r_{\mathrm{CH}})$ *maps $x \in \{0,1\}^*$ to $y \in \mathcal{Y}$ with help of the evaluation key ek_{CH} and a randomness $r_{\mathrm{CH}} \leftarrow \mathcal{R}_{\mathrm{CH}}$. If r_{CH} is uniformly distributed over $\mathcal{R}_{\mathrm{CH}}$, so is y over $\mathcal{Y}$.*

Equivocation. $\mathsf{CH.Equiv}(td_{\mathrm{CH}}, x, r_{\mathrm{CH}}, x')$ *outputs a randomness $r'_{\mathrm{CH}} \in \mathcal{R}_{\mathrm{CH}}$ such that*

$$\mathsf{CH.Eval}(ek_{\mathrm{CH}}, x; r_{\mathrm{CH}}) = \mathsf{CH.Eval}(ek_{\mathrm{CH}}, x'; r'_{\mathrm{CH}}), \tag{1}$$

for all x, x' and r_{CH}. Meanwhile, r'_{CH} is uniformly distributed as long as r_{CH} is.

Collision Resistance. *Given evaluation key ek_{CH}, it is hard to find $(x, r_{\mathrm{CH}}) \neq (x', r'_{\mathrm{CH}})$ with* $\mathsf{CH.Eval}(ek_{\mathrm{CH}}, x; r_{\mathrm{CH}}) = \mathsf{CH.Eval}(ek_{\mathrm{CH}}, x'; r'_{\mathrm{CH}})$. *More precisely, for any PPT adversary $\mathcal{A}$, the following advantage is negligible in κ.*

$$\mathsf{Adv}^{\mathrm{cr}}_{\mathsf{CH},\mathcal{A}}(\kappa) := \Pr\left[\begin{array}{c}(x, r_{\mathrm{CH}}) \neq (x', r'_{\mathrm{CH}}) \\ \wedge\ Eq.\ (1) holds.\end{array} : \begin{array}{l}(ek_{\mathrm{CH}}, td_{\mathrm{CH}}) \leftarrow \mathsf{CH.Gen}(1^\kappa) \\ (x, r_{\mathrm{CH}}, x', r'_{\mathrm{CH}}) \leftarrow \mathcal{A}(ek_{\mathrm{CH}})\end{array}\right]$$

3 The Construction

Let $\mathsf{HPS} = (\mathsf{HPS.Gen}, \mathsf{HPS.Pub}, \mathsf{HPS.Priv})$ be an ϵ_1-universal hash proof system, where $\mathsf{HPS.Gen}(1^\kappa)$ generates instances of params=(group, $\mathcal{K}$, $\mathcal{C}$, $\mathcal{V}$, $\mathcal{SK}$, $\mathcal{PK}$, $\Lambda_{(\cdot)} : \mathcal{C} \to \mathcal{K}$, $\mu : \mathcal{SK} \to \mathcal{PK}$). Let $\mathsf{LF} = (\mathsf{LF.Gen}, \mathsf{LF.Eval}, \mathsf{LF.LTag})$ be a $(\mathcal{K}, \ell_{\mathsf{LF}})$-one-time lossy filter. Define $\nu := \log(1/\epsilon_1)$. Let λ be a bound on the amount of leakage, and let $\mathsf{Ext} : \mathcal{K} \times \{0,1\}^d \to \{0,1\}^m$ be an average-case $(\nu - \lambda - \ell_{\mathsf{LF}}, \epsilon_2)$-strong extractor. We assume that ϵ_2 is negligible in κ. The encryption scheme $\mathsf{PKE} = (\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ with plaintext space $\{0,1\}^m$ is described as follows.

[1] In some case, a tag may neither injective nor lossy.

Key Generation. $\mathsf{PKE.Gen}(1^\kappa)$ runs $\mathsf{HPS.Gen}(1^\kappa)$ to obtain params and runs $\mathsf{LF.Gen}(1^\kappa)$ to obtain (Fpk, Ftd). It also picks $sk \leftarrow \mathcal{SK}$ and sets $pk = \mu(sk)$. The output is a public/secret key pair (PK, SK), where $PK = (\mathsf{params}, Fpk, pk)$ and $SK = sk$.

Encryption. $\mathsf{PKE.Enc}(PK, M)$ takes as input a public key PK and a message $M \in \{0,1\}^m$. It chooses $C \leftarrow \mathcal{V}$ with witness w, a random seed $s \leftarrow \{0,1\}^d$ and a random core tag $t_c \leftarrow \mathcal{T}_c$. It then computes

$$K = \mathsf{HPS.Pub}(pk, C, w),\ \ \Psi = \mathsf{Ext}(K, s) \oplus M,\ \ \Pi = \mathsf{LF}_{Fpk,t}(K),$$

where the filter tag is $t = (t_a, t_c)$ with $t_a = (C, s, \Psi)$. Output the ciphertext $CT = (C, s, \Psi, \Pi, t_c)$.

Decryption. $\mathsf{PKE.Dec}(SK, CT)$, given a secret key $SK = sk$ and a ciphertext $CT = (C, s, \Psi, \Pi, t_c)$, computes $K' = \mathsf{HPS.Priv}(sk, C)$ and $\Pi' = \mathsf{LF}_{Fpk,t}(K')$, where $t = ((C, s, \Psi), t_c)$. It checks whether $\Pi = \Pi'$. If not, it rejects with $\perp$. Otherwise it outputs $M = \Psi \oplus \mathsf{Ext}(K', s)$.

The correctness of PKE follows from the correctness of the underlying hash proof system.

The idea of our construction is to employ a Hash Proof System (HPS) to generate an encapsulated key K, which is then used not only to mask the plaintext, but also to verify the well-formedness of the ciphertext. To deal with the secret key leakage, an extractor converts K to a shorter key to hide the plaintext M. A one-time lossy filter $\mathsf{LF}_{Fpk,t}(K)$ helps to implement the verification. The filter in the challenge ciphertext CT^* works in the lossy mode, and it leaks only a limited amount of information about the key K. For any invalid ciphertext submitted by the adversary to the decryption oracle, the filter works in the injective mode with overwhelming probability. Consequently, the output of the filter in the invalid ciphertext preserves the entropy of K, which makes the ciphertext rejected by the decryption oracle with overwhelming probability.

The security of the construction is established by the theorem below.

Theorem 1. *Assuming that* HPS *is an ϵ_1-universal hash proof system,* LF *is a $(\mathcal{K}, \ell_{\mathsf{LF}})$-one-time lossy filter, and* $\mathsf{Ext} : \mathcal{K} \times \{0,1\}^d \to \{0,1\}^m$ *is an average-case $(\nu - \lambda - \ell_{\mathsf{LF}}, \epsilon_2)$-strong extractor, the encryption scheme* PKE *is λ-LR-CCA-secure as long as $\lambda \leq \nu - m - \ell_{\mathsf{LF}} - \omega(\log \kappa)$, where m is the plaintext length and $\nu := \log(1/\epsilon_1)$. Particularly,*

$$\mathsf{Adv}^{\text{lr-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\text{ind}}_{\mathsf{LF},\mathcal{B}_1}(\kappa) + Q(\kappa) \cdot \mathsf{Adv}^{\text{eva}}_{\mathsf{LF},\mathcal{B}_2}(\kappa) + \mathsf{Adv}^{\text{smp}}_{\mathsf{HPS},\mathcal{B}_3}(\kappa) + \frac{Q(\kappa) 2^{\lambda + \ell_{\mathsf{LF}} + m}}{2^\nu - Q(\kappa)} + \epsilon_2$$

where $Q(\kappa)$ denotes the number of decryption queries made by $\mathcal{A}$.

Parameters and leakage rate. To make our construction tolerate leakage as much as possible, it is useful to consider a "very strong" hash proof system (i.e., $\epsilon_1 \leq 2/|\mathcal{K}|$). In this case, $\nu = \log(1/\epsilon_1) \geq \log|\mathcal{K}| - 1$. Thus, when $\mathcal{K}$ is sufficiently large, the leakage rate (defined as $\lambda/|SK|$) in our construction approaches $(\log|\mathcal{K}|)/|SK|$ asymptotically.

CCA-security. Clearly, if $\lambda = 0$ and $\log(1/\epsilon_1) \geq m + \ell_{\mathsf{LF}} + \omega(\log \kappa)$, the above construction is CCA-secure. Thus, it provides a new approach for constructing CCA-secure PKE from any universal hash proof system and OT-LF.

Proof. The proof goes with game arguments [33]. We define a sequence of games, $\mathsf{Game}_0, \ldots, \mathsf{Game}_6$, played between a simulator Sim and a PPT adversary $\mathcal{A}$. In each game, the adversary outputs a bit b' as a guess of the random bit b used by the simulator. Denote by S_i the event that $b = b'$ in Game_i and denote by $CT^* = (C^*, s^*, \Psi^*, \Pi^*, t_c^*)$ the challenge ciphertext.

Game_0**:** This is the original LR-CCA game $\mathsf{Exp}^{\text{lr-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)$. The simulator generates the public/secret key pair (PK, SK) by invoking $\mathsf{PKE.Gen}(1^\kappa)$ and sends the public key PK to the adversary $\mathcal{A}$. For each decryption query CT or leakage query f_i, Sim responds with $\mathsf{PKE.Dec}(SK, CT)$ or $f_i(SK)$ using secret key SK. Upon receiving two messages M_0, M_1 of equal length from the adversary, Sim selects a random $b \in \{0,1\}$ and sends the challenge ciphertext $CT^* := \mathsf{PKE.Enc}(PK, M_b)$ to $\mathcal{A}$. The simulator continues to answer the adversary's decryption query as long as $CT \neq CT^*$. Finally, $\mathcal{A}$ outputs a bit b', which is a guess of b. By the Definition 2, we have $\mathsf{Adv}^{\text{lr-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa) := \left|\Pr[S_0] - \frac{1}{2}\right|$.

Game_1**:** This game is exactly like Game_0, except for $\mathsf{PKE.Gen}(1^\kappa)$ and the generation of the core tag t_c^* of the filter tag in the challenge ciphertext. When calling $\mathsf{PKE.Gen}(1^\kappa)$, the simulator keeps the trapdoor Ftd of LF as well as SK. Instead of sampling t_c^* at random from $\mathcal{T}_c$, Sim computes t_c^* with $\mathsf{LF.LTag}(Ftd, t_a^*)$, where $t_a^* = (C^*, s^*, \Psi^*)$. A straightforward reduction to LF's indistinguishability of lossy tag and random tag yields $|\Pr[S_1] - \Pr[S_0]| \leq \mathsf{Adv}^{\mathrm{ind}}_{\mathsf{LF},\mathcal{B}_1}(\kappa)$ for a suitable adversary $\mathcal{B}_1$ on LF's indistinguishability.

Game_2**:** This game is exactly like Game_1, except that a special rejection rule applies to the decryption oracle. If the adversary queries a ciphertext $CT = (C, s, \Psi, \Pi, t_c)$ such that $t = (t_a, t_c) = (t_a^*, t_c^*) = t^*$, then the decryption oracle immediately outputs $\perp$ and halts. For convenient, we call such tag a copied LF tag. We show that a decryption query with a copied LF tag is rejected in decryption oracles in both Game_1 and Game_2. We consider the following two cases.

- case 1: $\Pi = \Pi^*$. This implies $CT = CT^*$. In this case the decryption oracles in Game_1 and Game_2 proceed identically since $\mathcal{A}$ is not allowed to ask for the decryption of challenge ciphertext.
- case 2: $\Pi \neq \Pi^*$. Since $t = ((C, s, \Psi), t_c) = ((C^*, s^*, \Psi^*), t_c^*) = t^*$, it follows that $K = K^*$, and thus $\mathsf{LF}_{Fpk,t}(K) = \mathsf{LF}_{Fpk,t^*}(K^*) = \Pi^*$. So, such decryption queries would have been rejected already in Game_1.

According to above analysis, we have $\Pr[S_2] = \Pr[S_1]$.

Game_3**:** This game is exactly like Game_2, except for the generation of K^* used in the challenge ciphertext. In this game, Sim computes K^* with $\mathsf{HPS.Priv}(sk, C^*)$ instead of $\mathsf{HPS.Pub}(pk, C^*, w^*)$. Since HPS is projective, this change is purely conceptual, and thus $\Pr[S_3] = \Pr[S_2]$.

Game_4**:** This game is exactly like Game_3, except for the generation of C^* in the challenge ciphertext $CT^* = (C^*, s^*, \Psi^*, \Pi^*, t_c^*)$. Now Sim samples C^* from $\mathcal{C} \setminus \mathcal{V}$ instead of $\mathcal{V}$. A straightforward reduction to the indistinguishability of the subset membership problem yields $|\Pr[S_4] - \Pr[S_3]| \leq \mathsf{Adv}^{\mathrm{smp}}_{\mathsf{HPS},\mathcal{B}_3}(\kappa)$ for a suitable adversary $\mathcal{B}_3$.

Game_5**:** This game is the same as Game_4, except that another special rejection rule is applied to the decryption oracle. If the adversary queries a ciphertext $CT = (C, s, \Psi, \Pi, t_c)$ for decryption such that $C \in \mathcal{C} \setminus \mathcal{V}$, then the decryption oracle immediately outputs $\perp$. Let bad_C be the event that a ciphertext is rejected in Game_5 that would not have been rejected under the rules of Game_4. Then Game_5 and Game_4 proceed identically until event bad_C occurs. We have

$$|\Pr[S_5] - \Pr[S_4]| \leq \Pr[\mathsf{bad}_C] \tag{2}$$

by the difference lemma of [33]. We show the following lemma shortly (after the main proof), which guarantees that bad_C occurs with a negligible probability.

Lemma 3. *Suppose that the adversary $\mathcal{A}$ makes at most $Q(\kappa)$ decryption queries. Then*

$$\Pr[\mathsf{bad}_C] \leq Q(\kappa) \cdot \mathsf{Adv}^{\mathrm{eva}}_{\mathsf{LF},\mathcal{B}}(\kappa) + \frac{Q(\kappa) 2^{\lambda + \ell_{\mathsf{LF}} + m}}{2^\nu - Q(\kappa)} \tag{3}$$

where $\mathcal{B}$ is a suitable adversary attacking on LF's evasiveness.

Game_6**:** This game is exactly like Game_5, except for the generation of Ψ^* in CT^*. In this game, Sim chooses Ψ^* uniformly at random from $\{0,1\}^m$ instead of using $\mathsf{Ext}(\Lambda_{sk}(C^*), s^*) \oplus M_b$.

Claim 1. *For $C^* \leftarrow \mathcal{C} \setminus \mathcal{V}$ if the decryption algorithm rejects all invalid ciphertexts, then the value $\Lambda_{sk}(C^*)$ has average min-entropy at least $\nu - \lambda - \ell_{\mathsf{LF}} \geq \omega(\log \kappa) + m$ given all the other values in $\mathcal{A}$'s view (denoted by $\mathsf{view}'_{\mathcal{A}}$).*

We prove Claim 1 by directly analyzing the average min-entropy of $\Lambda_{sk}(C^*)$ from the adversary's point of view. Since all invalid ciphertexts are rejected by the decryption oracle in both Game_5 and Game_6, $\mathcal{A}$ cannot learn more information on the value $\Lambda_{sk}(C^*)$ from the decryption oracle other than pk, C^*, Π^* and the key leakage. Recall that Π^* has only $2^{\ell_{\mathsf{LF}}}$ possible vales and $\mathrm{H}_\infty(\Lambda_{sk}(C^*) \mid (pk, C^*)) \geq \nu$ (which holds for all pk and $C^* \in \mathcal{C} \setminus \mathcal{V}$). Hence,

$$\begin{aligned}\widetilde{\mathrm{H}}_\infty(\Lambda_{sk}(C^*) \mid \mathsf{view}'_{\mathcal{A}}) &= \widetilde{\mathrm{H}}_\infty(\Lambda_{sk}(C^*) \mid pk, C^*, \lambda\text{-leakage}, \Pi^*) \\ &\geq \widetilde{\mathrm{H}}_\infty(\Lambda_{sk}(C^*) \mid pk, C^*) - \lambda - \ell_{\mathsf{LF}} \geq \nu - \lambda - \ell_{\mathsf{LF}}\end{aligned}$$

according to Lemma 1.

Applying an average-case $(\nu - \lambda - \ell_{\mathsf{LF}}, \epsilon_2)$-strong extractor $\mathsf{Ext} : \mathcal{K} \times \{0,1\}^d \to \{0,1\}^m$ to $\Lambda_{sk}(C^*)$, we have that $\mathsf{Ext}(\Lambda_{sk}(C^*), s^*)$ is ϵ_2-close to uniform given $\mathcal{A}$'s view. Hence,

$$|\Pr[S_6] - \Pr[S_5]| \leq \epsilon_2 \tag{4}$$

Observe that in Game_6, the challenge ciphertext is completely independent of the random coin b picked by the simulator. Thus, $\Pr[S_6] = 1/2$.

Putting all together, Theorem 1 follows. □

It remains to prove Lemma 3. We do it now.

Proof of Lemma 3. Let F be the event that in Game_4 there exists a decryption query $CT = (C, s, \Psi, \Pi, t_c)$, such that $t = ((C, s, \Psi), t_c)$ is a non-injective, non-copied tag. We have

$$\Pr[\mathsf{bad}_C] = \Pr[\mathsf{bad}_C \wedge F] + \Pr[\mathsf{bad}_C \wedge \overline{F}] \leq \Pr[F] + \Pr[\mathsf{bad}_C \mid \overline{F}] \tag{5}$$

Thus, it suffices to prove the following two claims: Claim 2 and Claim 3.

Claim 2. *Suppose that the adversary $\mathcal{A}$ makes at most $Q(\kappa)$ decryption queries. If LF is a one-time lossy filter, then*

$$\Pr[F] \leq Q(\kappa) \cdot \mathsf{Adv}^{\mathrm{eva}}_{\mathsf{LF},\mathcal{B}}(\kappa) \tag{6}$$

where $\mathcal{B}$ is a suitable adversary on LF's evasiveness.

Proof. Given a challenge LF evaluation key F^*_{pk}, $\mathcal{B}$ simulates $\mathcal{A}$'s environment in Game_4 as follows. It generates the PKE's public key PK as in Game_4 but sets $F_{pk} = F^*_{pk}$. Note that $\mathcal{B}$ can use PKE's secret key to deal with $\mathcal{A}$'s decryption queries. To simulate the challenge ciphertext (in which the LF tag should be lossy), $\mathcal{B}$ queries its lossy tag generation oracle once with $t^*_a = (C^*, s^*, \Psi^*)$ to proceed t^*_c, where (C^*, s^*, Ψ^*) are generated as in Game_4. Finally, $\mathcal{B}$ chooses $i \in [Q(k)]$ uniformly, and outputs the tag $t = ((C, s, \Psi), t_c)$ extracted from $\mathcal{A}$'s i-th decryption query (C, s, Ψ, Π, t_c). Clearly, if the event F occurs, with probability at least $1/Q(\kappa)$, t is a non-injective tag. That is $\Pr[F] \leq Q(\kappa) \cdot \mathsf{Adv}^{\mathrm{eva}}_{\mathsf{LF},\mathcal{B}}(\kappa)$. □

Claim 3. *Suppose that the adversary $\mathcal{A}$ makes at most $Q(\kappa)$ decryption queries. If HPS is ϵ_1-universal, then*

$$\Pr[\mathsf{bad}_C \mid \overline{F}] \leq \frac{Q(\kappa) 2^{\lambda + \ell_{\mathsf{LF}} + m}}{2^\nu - Q(\kappa)} \tag{7}$$

where $\nu = \log(1/\epsilon_1)$.

Proof. Suppose that $CT = (C, s, \Psi, \Pi, t_c)$ is the first ciphertext that makes bad_C happen given $\overline{F}$, i.e. $C \in \mathcal{C} \setminus \mathcal{V}$ but $\Pi = \mathsf{LF}_{Fpk,t}(\Lambda_{sk}(C))$, where $t = ((C, s, \Psi), t_c)$ is an injective LF tag. For simplicity, we call $CT = (C, s, \Psi, \Pi, t_c)$ an *invalid* ciphertext if $C \in \mathcal{C} \setminus \mathcal{V}$. Denote by $\mathsf{view}_{\mathcal{A}}$ the adversary's view prior to submitting the first invalid ciphertext. Observe that only pk, the challenge ciphertext CT^*, and the key leakage of at most λ bits reveal information of the secret key to the adversary. According to Lemma 1, we have

$$\begin{aligned}\widetilde{\mathrm{H}}_\infty(\Lambda_{sk}(C) \mid \mathsf{view}_\mathcal{A}) &= \widetilde{\mathrm{H}}_\infty(\Lambda_{sk}(C) \mid pk, C, CT^*, \lambda\text{-leakage}) \\ &\geq \widetilde{\mathrm{H}}_\infty(\Lambda_{sk}(C) \mid pk, C, CT^*) - \lambda \\ &\geq \mathrm{H}_\infty(\Lambda_{sk}(C) \mid (pk, C)) - \lambda - \ell_{\mathsf{LF}} - m \quad (8) \\ &\geq \nu - \lambda - \ell_{\mathsf{LF}} - m \quad (9)\end{aligned}$$

Eq. (8) follows from the fact that in the challenge ciphertext CT^*, only Ψ^* and Π^* are related to the secret key, and Ψ^* has at most 2^m possible values and Π^* has at most $2^{\ell_{\mathsf{LF}}}$ possible values. Note that the information revealed by t_c^* has already been completely taken into account by Ψ^*, since $t_c^* = \mathsf{LF.LTag}(Ftd, (C^*, s^*, \Psi^*))$ can be regarded as a function of Ψ^*. Eq. (9) follows from the fact that for all pk and $C \in \mathcal{C} \setminus \mathcal{V}$, $\mathrm{H}_\infty(\Lambda_{sk}(C) \mid (pk, C)) \geq \log(1/\epsilon_1) = \nu$, which is due to the ϵ_1-universal property of HPS and Lemma 2. The fact that event F does not occur implies that $t = ((C, s, \Psi), t_c)$ is an injective tag. Applying an injective function to a distribution preserves its min-entropy, we have $\widetilde{\mathrm{H}}_\infty(\mathsf{LF}_{Fpk,t}(\Lambda_{sk}(C)) \mid \mathsf{view}_\mathcal{A}) \geq \nu - \lambda - \ell_{\mathsf{LF}} - m$. Thus, in Game_4 the decryption algorithm accepts the first invalid ciphertext with probability at most $2^{\lambda+\ell_{\mathsf{LF}}+m}/2^\nu$. Observe that the adversary can rule out one more value of K from each rejection of invalid ciphertext. So, the decryption algorithm accepts the i-th invalid ciphertext with probability at most $2^{\lambda+\ell_{\mathsf{LF}}+m}/(2^\nu - i + 1)$. Since $\mathcal{A}$ makes at most $Q(\kappa)$ decryption queries, it follows that

$$\Pr[\mathsf{bad}_C \mid \overline{F}] \leq \frac{Q(\kappa)2^{\lambda+\ell_{\mathsf{LF}}+m}}{2^\nu - Q(\kappa)} \quad (10)$$

which is negligible in κ if $\lambda \leq \nu - m - \ell_{\mathsf{LF}} - \omega(\log \kappa)$. □

This completes the proof of Lemma 3. □

4 Instantiation from the DDH Assumption

This section is organized as follows. In Section 4.1, we present a variant of hash proof system from the Decisional Diffie-Hellman (DDH) assumption [9]. In Section 4.2, we introduce an efficient DDH-based OT-LF. In Section 4.3, we apply the construction in Section 3 to the two building blocks and obtain an efficient DDH-based LR-CCA secure PKE scheme, depicted in Fig. 1. In Section 4.4, we show a comparison of our scheme with some existing LR-CCA secure PKE schemes.

The DDH Assumption. We assume a PPT algorithm $\mathcal{G}(1^\kappa)$ that takes as input 1^κ and outputs a tuple of $\mathbb{G} = \langle q, G, g\rangle$, where G is a cyclic group of prime order q and g is a generator of G. The Decisional Diffie-Hellman (DDH) assumption holds iff

$$\mathsf{Adv}^{\mathrm{ddh}}_{G,\mathcal{D}}(\kappa) = \left|\Pr[\mathcal{D}(g_1, g_2, g_1^r, g_2^r) = 1] - \Pr[\mathcal{D}(g_1, g_2, g_1^r, g_2^{r'}) = 1]\right|$$

is negligible in κ for any PPT adversary $\mathcal{D}$, where $g_1, g_2 \leftarrow G$, $r \leftarrow \mathbb{Z}_q$ and $r' \leftarrow \mathbb{Z}_q \setminus \{r\}$.

4.1 A DDH-based HPS

Let $\langle q, G, g\rangle \leftarrow \mathcal{G}(1^\kappa)$ and let g_1, g_2 be two random generators of G. Choose $\mathfrak{n} \in \mathbb{N}$. We assume there is an efficient injective mapping $\mathsf{Inj} : G \to \mathbb{Z}_q$ [2]. For any $u = (u_1, \ldots, u_\mathfrak{n}) \in G^\mathfrak{n}$, let $\widetilde{\mathsf{Inj}}(u) = (\mathsf{Inj}(u_1), \ldots, \mathsf{Inj}(u_\mathfrak{n})) \in \mathbb{Z}_q^\mathfrak{n}$. Clearly, $\widetilde{\mathsf{Inj}}$ is also an injection. We define a hash proof system $\mathsf{HPS}_1 = (\mathsf{HPS}_1.\mathsf{Gen}, \mathsf{HPS}_1.\mathsf{Pub}, \mathsf{HPS}_1.\mathsf{Priv})$ below.

The parameter $\mathsf{params} = (\mathsf{group}, \mathcal{K}, \mathcal{C}, \mathcal{V}, \mathcal{SK}, \mathcal{PK}, \Lambda_{sk}, \mu)$ is set up as follows.

[2]For example, G is a q-order elliptic curve group over finite field $\mathbb{F}_p$. For 80-bit security, p and q can be chosen to be 160-bit primes. In such a group, elements (i.e., elliptic curve points) can be represented by 160-bit strings.

- $\mathsf{group} = \langle q, G, g_1, g_2, \mathfrak{n}\rangle$, $\mathcal{C} = G \times G$, $\mathcal{V} = \{(g_1^r, g_2^r) : r \in \mathbb{Z}_q\}$ with witness set $W = \mathbb{Z}_q$.
- $\mathcal{K} = \mathbb{Z}_q^{\mathfrak{n}}$, $\mathcal{SK} = (\mathbb{Z}_q \times \mathbb{Z}_q)^{\mathfrak{n}}$, $\mathcal{PK} = G^{\mathfrak{n}}$.
- For $sk = (x_{i,1}, x_{i,2})_{i\in[\mathfrak{n}]} \in \mathcal{SK}$, define $pk = (pk_i)_{i\in[\mathfrak{n}]} = \mu(sk) = (g_1^{x_{i,1}} g_2^{x_{i,2}})_{i\in[\mathfrak{n}]}$.
- For all $C = (u_1, u_2) \in \mathcal{C}$, define $\Lambda_{sk}(C) = \widetilde{\mathsf{Inj}}((u_1^{x_{i,1}} u_2^{x_{i,2}})_{i\in[\mathfrak{n}]})$.

The public evaluation and private evaluation algorithms are defined as follows:

- For all $C = (g_1^r, g_2^r) \in \mathcal{V}$ with witness $r \in \mathbb{Z}_q$, define $\mathsf{HPS_1.Pub}(pk, C, r) = \widetilde{\mathsf{Inj}}(pk_1^r, \ldots, pk_{\mathfrak{n}}^r)$.
- For all $C = (u_1, u_2) \in \mathcal{C}$, define $\mathsf{HPS_1.Priv}(sk, C) = \Lambda_{sk}(C)$.

Correctness of $\mathsf{HPS_1}$ follows directly by the definitions of μ and Λ_{sk}. The subset membership problem in $\mathsf{HPS_1}$ is hard because of the DDH assumption. If $\mathfrak{n} = 1$, this is just the DDH-based hash proof system introduced by Cramer and Shoup with encapsulated key set $\mathcal{K} = \mathbb{Z}_q$, and is known to be perfectly universal [9, 21]. We have the following theorem with proof in Appendix A.

Theorem 2. *For any $\mathfrak{n} \in \mathbb{N}$, $\mathsf{HPS_1}$ is perfectly universal under the DDH assumption with encapsulated key size $|\mathcal{K}| = q^{\mathfrak{n}}$.*

4.2 A DDH-based OT-LF

We use the following notations. If $A = (A_{i,j})$ is an $n \times n$ matrix over $\mathbb{Z}_{\widetilde{q}}$, and $\widetilde{g}$ is an element of $\widetilde{q}$-order group $\widetilde{G}$. Then $\widetilde{g}^A$ denotes the $n \times n$ matrix $(\widetilde{g}^{A_{i,j}})$ over $\widetilde{G}$. Given a vector $X = (X_1, \ldots, X_n) \in \mathbb{Z}_{\widetilde{q}}^n$ and an $n \times n$ matrix $E = (E_{i,j}) \in \widetilde{G}^{n\times n}$, define

$$X \cdot E := (\prod_{i=1}^{n} E_{i,1}^{X_i}, \ldots, \prod_{i=1}^{n} E_{i,n}^{X_i}) \in \widetilde{G}^n.$$

Let $\mathsf{CH} = (\mathsf{CH.Gen}, \mathsf{CH.Eval}, \mathsf{CH.Equiv})$ define a chameleon hashing function with image set $\mathbb{Z}_{\widetilde{q}}$. The OT-LF is $\mathsf{LF_1} = (\mathsf{LF_1.Gen}, \mathsf{LF_1.Eval}, \mathsf{LF_1.LTag})$, as shown below.

Key Generation. $\mathsf{LF_1.Gen}(1^\kappa)$ runs $\mathcal{G}(1^\kappa)$ to obtain $\widetilde{\mathbb{G}} = \langle \widetilde{q}, \widetilde{G}, \widetilde{g}\rangle$ and runs $\mathsf{CH.Gen}(1^\kappa)$ to obtain $(ek_{\text{CH}}, td_{\text{CH}})$. Pick a random pair $(t_a^*, t_c^*) \leftarrow \{0,1\}^* \times \mathcal{R}_{\text{CH}}$ and compute $b^* = \mathsf{CH.Eval}(ek_{\text{CH}}, t_a^*; t_c^*)$. Choose $r_1, \ldots, r_n, s_1, \ldots, s_n \leftarrow \mathbb{Z}_{\widetilde{q}}$, and compute an $n\times n$ matrix $A = (A_{i,j}) \in \mathbb{Z}_{\widetilde{q}}^{n\times n}$ with $A_{i,j} = r_i s_j$ for $i, j \in [n]$. Compute matrix $E = \widetilde{g}^{A-b^*\mathbf{I}} \in \widetilde{G}^{n\times n}$, where $\mathbf{I}$ is the $n \times n$ identity matrix over $\mathbb{Z}_{\widetilde{q}}$. Finally, output $Fpk = (\widetilde{q}, \widetilde{G}, \widetilde{g}, ek_{\text{CH}}, E)$ and $Ftd = (td_{\text{CH}}, t_a^*, t_c^*)$. The tag space is defined as $\mathcal{T} = \{0,1\}^* \times \mathcal{R}_{\text{CH}}$, where $\mathcal{T}_{loss} = \{(t_a, t_c) : (t_a, t_c) \in \mathcal{T} \wedge \mathsf{CH.Eval}(ek_{\text{CH}}, t_a; t_c) = b^*\}$ and $\mathcal{T}_{inj} = \{(t_a, t_c) : (t_a, t_c) \in \mathcal{T} \wedge \mathsf{CH.Eval}(ek_{\text{CH}}, t_a; t_c) \notin \{b^*, b^* - \mathbf{Tr}(A)\}\}$.

Evaluation. For a tag $t = (t_a, t_c) \in \{0,1\}^* \times \mathcal{R}_{\text{CH}}$ and an input $X = (X_1, \ldots, X_n) \in \mathbb{Z}_{\widetilde{q}}^n$, $\mathsf{LF_1.Eval}(Fpk, t, X)$ first computes $b = \mathsf{CH.Eval}(ek_{\text{CH}}, t_a; t_c)$ and outputs

$$y = X \cdot (E \otimes \widetilde{g}^{b\mathbf{I}}),$$

where "$\otimes$" denotes the operation of entry-wise multiplication.

Lossy Tag Generation. For an auxiliary tag t_a, $\mathsf{LF_1.LTag}(Ftd, t_a)$ computes a core tag $t_c = \mathsf{CH.Equiv}(td_{\text{CH}}, t_a^*, t_c^*, t_a)$ with the trapdoor $Ftd = (td_{\text{CH}}, t_a^*, t_c^*)$.

Theorem 3. $\mathsf{LF_1}$ *is a $(\mathbb{Z}_{\widetilde{q}}^n, \log \widetilde{q})$-OT-LF under the DDH assumption.*

Proof. The proof of Theorem 3 is given in Appendix B. □

4.3 The DDH-based PKE Scheme

Let $\mathbb{G} = \langle q, G, g\rangle$ and $\widetilde{\mathbb{G}} = \langle \widetilde{q}, \widetilde{G}, \widetilde{g}\rangle$ be two group descriptions. Suppose $\mathfrak{n} \in \mathbb{N}$ satisfies $\mathfrak{n}\log q \geq \log\widetilde{q} + \lambda + m + \omega(\log\kappa)$. Set $n = \lceil \mathfrak{n}\log q/\log\widetilde{q}\rceil$. Let $(ek_{\text{CH}}, td_{\text{CH}}) \leftarrow \mathsf{CH.Gen}(1^\kappa)$ be a chameleon hash function with image set $\mathbb{Z}_{\widetilde{q}}$. Let $\mathsf{Ext} : \mathbb{Z}_q^{\mathfrak{n}} \times \{0,1\}^d \to \{0,1\}^m$ be an average-case $(\mathfrak{n}\log q - \log\widetilde{q} - \lambda, \epsilon_2)$-strong extractor. Applying the general construction in Section 3 to the aforementioned DDH-based HPS and OT-LF, we obtain a DDH-based PKE scheme in Fig. 1.

Key Generation. $\mathsf{PKE}_1.\mathsf{Gen}(1^\kappa)$: Choose $g_1, g_2 \leftarrow G$ and $(x_{i,1}, x_{i,2}) \leftarrow \mathbb{Z}_q$ for $i \in [\mathfrak{n}]$. Set $pk_i = g_1^{x_{i,1}} g_2^{x_{i,2}}$ for $i \in [\mathfrak{n}]$. Also choose a random pair $(t_a^*, t_c^*) \in \{0,1\}^* \times \mathcal{R}_{\text{CH}}$ and set $b^* = \mathsf{CH.Eval}(ek_{\text{CH}}, t_a^*; t_c^*)$. Choose $r_1, \ldots, r_n,\ s_1, \ldots, s_n \leftarrow \mathbb{Z}_{\widetilde{q}}$, and compute matrix $E = (E_{i,j})_{i,j\in[n]} \in \widetilde{\mathbb{G}}^{n\times n}$, where $E_{i,j} = \widetilde{g}^{r_i s_j}$ for $i, j \in [n], i \neq j$, and $E_{i,i} = \widetilde{g}^{r_i s_i}\widetilde{g}^{-b^*}$ for $i \in [n]$. Return $PK = (q, G, g_1, g_2, \mathfrak{n}, (pk_i)_{i\in[\mathfrak{n}]}, \widetilde{q}, \widetilde{G}, \widetilde{g}, E, ek_{\text{CH}})$ and $SK = (x_{i,1}, x_{i,2})_{i\in[\mathfrak{n}]}$.

Encryption. $\mathsf{PKE}_1.\mathsf{Enc}(PK, M)$: For a public key PK and a message $M \in \{0,1\}^m$, it chooses $r \leftarrow \mathbb{Z}_q$ and $s \leftarrow \{0,1\}^d$. Compute

$$C = (g_1^r, g_2^r),\quad K = \widetilde{\mathsf{Inj}}\left(pk_1^r, \ldots, pk_{\mathfrak{n}}^r\right),\quad \Psi = \mathsf{Ext}(K, s) \oplus M,\quad \Pi = K \cdot (E \otimes \widetilde{g}^{b\mathbf{I}})$$

where $b = \mathsf{CH.Eval}(ek_{\text{CH}}, t_a; t_c)$ for the auxiliary tag $t_a = (C, s, \Psi)$ and a random filter core tag $t_c \in \mathcal{R}_{\text{CH}}$. Note that in the computation of Π, K is regarded as a vector of dimension n over $\mathbb{Z}_{\widetilde{q}}$ (this works well since $\mathfrak{n}\log q \leq n\log\widetilde{q}$). Return $CT = (C, s, \Psi, \Pi, t_c) \in G^2 \times \{0,1\}^d \times \{0,1\}^m \times \widetilde{G}^n \times \mathcal{R}_{\text{CH}}$.

Decryption. $\mathsf{PKE}_1.\mathsf{Dec}(SK, CT)$: For a ciphertext $CT = (C, s, \Psi, \Pi, t_c)$, it parses C as $(u_1, u_2) \in G^2$ and then computes $K' = \widetilde{\mathsf{Inj}}\left(u_1^{x_{1,1}} u_2^{x_{1,2}}, \ldots, u_1^{x_{\mathfrak{n},1}} u_2^{x_{\mathfrak{n},2}}\right)$ and $\Pi' = K' \cdot (E \otimes \widetilde{g}^{b\mathbf{I}})$, where $b = \mathsf{CH.Eval}(ek_{\text{CH}}, (C, s, \Psi); t_c)$. Finally, it checks whether $\Pi = \Pi'$. If not, it rejects with $\perp$. Else, it returns $M = \Psi \oplus \mathsf{Ext}(K', s)$.

Figure 1: A DDH-based PKE Scheme $\mathsf{PKE}_1 = (\mathsf{PKE}_1.\mathsf{Gen}, \mathsf{PKE}_1.\mathsf{Enc}, \mathsf{PKE}_1.\mathsf{Dec})$

Theorem 4. *If the DDH assumptions hold in groups G and $\widetilde{G}$, and the* CH *is a chameleon hash function, then* PKE_1 *is λ-LR-CCA secure if $\lambda \leq \mathfrak{n}\log q - \log\widetilde{q} - m - \omega(\log\kappa)$ (i.e., $\mathfrak{n} \geq (\lambda + \log\widetilde{q} + m + \omega(\log\kappa))/\log q$). In particular, the leakage rate in* PKE_1 *is $1/2 - o(1)$ and*

$$\begin{aligned}\mathsf{Adv}^{\text{lr-cca}}_{\mathsf{PKE}_1,\mathcal{A}}(\kappa) \leq \quad & \mathsf{Adv}^{\text{ddh}}_{G,\mathcal{B}_1}(\kappa) + 2\mathfrak{n}\mathsf{Adv}^{\text{ddh}}_{\widetilde{G},\mathcal{B}_2}(\kappa) + \frac{Q(\kappa)\cdot\widetilde{q}\cdot 2^{\lambda+m}}{q^{\mathfrak{n}} - Q(\kappa)} + \epsilon_2 \\ & + Q(\kappa)\left((2\mathfrak{n}+1)\mathsf{Adv}^{\text{ddh}}_{\widetilde{G},\mathcal{B}_2}(\kappa) + \mathsf{Adv}^{\text{cr}}_{\mathsf{CH},\mathcal{B}_3}(\kappa)\right)\end{aligned}$$

where $Q(\kappa)$ is the number of decryption queries made by $\mathcal{A}$.

Proof. Theorem 2 showed that the underlying HPS in PKE_1 is perfectly universal (i.e., $\epsilon_1 = 1/q^{\mathfrak{n}}$). Theorem 3 said that the underlying filter is a $(\widetilde{q}^n, \log\widetilde{q})$-OT-LF. Consequently, PKE_1 is λ-LR-CCA secure according to Theorem 1. If the parameter $\mathfrak{n}$ in PKE_1 increases, with $\widetilde{q}, m$ fixed, $\lambda/|SK| = (\mathfrak{n}\log q - \log\widetilde{q} - m - \omega(\log\kappa))/(2\mathfrak{n}\log q) = 1/2 - o(1)$. □

4.4 Efficiency Discussion

In this section, we show a comparison of our DDH-based PKE scheme with the existing DDH/DLIN based LR-CCA secure PKE schemes [28, 25, 11, 16] in terms of leakage rate and ciphertext overhead (defined as the difference between the ciphertext length and the embedded message length). Note that the GHV12 scheme is obtained by applying the CHK transformation to the mKDM-sID-CPA secure scheme

Table 1: Secret-key size, leakage amount and ciphertext overhead

Schemes	SK size (# bits)	Leakage amount (# bits)	CT overhead ($\#G$)
GHV12 [16]	$\mathfrak{n}\log q$	$\lambda \leq \mathfrak{n}\log q - 3\log q - 2\ell(\kappa)$	$2\mathfrak{n}+6$
NS09 [28]	$6\log q'$	$\lambda \leq \log q' - \omega(\log\kappa) - m$	3
LZSS12 [25]	$4\log q'$	$\lambda \leq \log q' - \omega(\log\kappa) - m$	3
Ours	$2\mathfrak{n}\log q$	$\lambda \leq \mathfrak{n}\log q - \log\widetilde{q} - m - \omega(\log\kappa)$	$\mathfrak{n}+2$

Table 2: Relations between ciphertext overhead and leakage rate

Schemes	CT overhead ($\#\ell(\kappa)$ bits)	Leakage rate interval (δ)	Assumption
DHLW10 [11]	$21/(1-\delta)+70$	$[0,1)$	DLIN (with tSE-NIZK)
GHV12 [16]	$4\lceil 4/(1-\delta)\rceil + 12$	$[0,1)$	DLIN (without proof)
NS09 [28]	$9/(1-6\delta)$	$[0,1/6)$	DDH
LZSS12 [25]	$9/(1-4\delta)$	$[0,1/4)$	DDH
Ours	$2\lceil 5/(2-4\delta)\rceil + 4$	$[0,1/2)$	DDH

[16]. The GHV12 scheme is LR-CCA secure only if the mKDM-sID-CPA secure scheme [16] is master-key leakage sID-CPA secure. In fact, Galindo et.al. claimed their mKDM-sID-CPA secure scheme is master-key leakage sID-CPA secure with leakage rate $1-o(1)$, but without any rigorous proof. We personally regard that proving that claim is very hard, since the proof involves constructing a PPT simulator to answer not only key leakage queries, but also identities' private key queries. Nevertheless, we include the GHV12 scheme in the comparison. For simplicity, in a ciphertext, we only consider the length of group elements, ignoring the constant length non-group elements, e.g., the seed used in a randomness extractor. We also assume that elements in q-order group can be encoded as bit strings of length $\log q$. To be fair, like in [11, Theorem 6], we will consider the ciphertext overhead (shorted as "CT overhead") under any fixed and achievable leakage rate. We begin by giving an overview of the secret key size (shorted as "SK size"), the amount of absolute leakage and the number of group elements in the ciphertexts of the PKE schemes [28, 25, 16] in Table 1. In table 1, κ is the security parameter; q', q and $\widetilde{q}$ are group sizes; m is the message length and $\mathfrak{n}$ is a parameter as in Fig. 1 and [16, Section 5]. In our scheme, $n = \lceil \mathfrak{n}\log q/\log\widetilde{q}\rceil$. So, the bit-length of n elements in group $\widetilde{G}$ equals that of $\mathfrak{n}$ elements in group G.

We observe that in our scheme as well as that of [11, 16] the group size (i.e. q and $\widetilde{q}$) remains constant even with larger leakage. While in [28] and [25], both of them rely on increasing the group size (i.e., q') to tolerate larger leakage. So, it is more reasonable to compare the bit-length of ciphertext overhead rather than the number of group elements for the same leakage rate. As an example, we give the concrete relations between ciphertext overhead and leakage-rate of our scheme. In our scheme, for a security level $\ell(\kappa)$, we can choose $|q| = |\widetilde{q}| = 2\ell(\kappa)$. From [13], applying a universal hash function to a source with $3\ell(\kappa)$ entropy suffices to extract $\ell(\kappa)$-bit random key that is $2^{-\ell(\kappa)}$-close to a uniform distribution over $\{0,1\}^{\ell(\kappa)}$. So, we can set $\omega(\log\kappa) = 2\ell(\kappa)$ and $m = \ell(\kappa)$. According to Theorem 4, the amount of leakage is bounded by $(2\mathfrak{n}-5)\ell(\kappa)$. Thus, for any $\delta \in [0,1/2)$, the leakage rate in our scheme achieves δ, as long as $\mathfrak{n} \geq \lceil 5/(2-4\delta)\rceil$ (i.e., $\lambda \leq \ell(\kappa)(2\lceil 5/(2-4\delta)\rceil - 5)$) and the ciphertext overhead is $(\lceil 5/(2-4\delta)\rceil + 2)2\ell(\kappa)$ bits (ignoring the seed and the core tag part).

Similarly, we can compute the other schemes' ciphertext overheads for reasonable leakage rates. We summarize these results in Table 2.

Finally, we give a quantitative comparison among these LR-CCA secure PKE schemes in Table 3. While for some achievable leakage rate (e.g., $\delta \leq 0.4$), our scheme is more efficient compared with the other four schemes. As our construction is general, we can also instantiate it under other standard assumptions, e.g., the DCR assumption [29, 10]. In [16], the scheme is obtained by applying the CHK transformation [6] to a master-key leakage resilient identity-based encryption scheme. To the best of our knowledge, the constructions of identity-based PKE schemes [16, 24] with master-key leakage-resilience

Table 3: Quantitative comparison (# $\ell(\kappa)$-bit)

Leakage-rate / Schemes	1/8	1/6	1/4	1/3	3/8	2/5	1/2	1
DHLW10 [11]	94	95.2	98	101.5	103.6	105	112	-
GHV12 [16]	32	32	36	36	40	40	44	-
NS12 [28]	36	-	-	-	-	-	-	-
LZSS12 [25]	18	27	-	-	-	-	-	-
Ours	**12**	**12**	**14**	**20**	**24**	**30**	-	-

are all based on the assumptions (e.g., DLIN) over bilinear groups. Our schemes are the first DDH/DCR based efficient LR-CCA secure PKE schemes with leakage rate $1/2 - o(1)$.

5 Instantiation from the DCR Assumption

Here we present a PKE scheme instantiated under the Decisional Composite Residuosity (DCR) assumption [29]. The underlying HPS and OT-LF are given in Appendix C and Appendix D.

Definition 7 (The DCR Assumption [29, 10])**.** *We assume a PPT algorithm $\mathcal{IG}(1^\kappa)$ that on input 1^κ generates a Blum integer $N = PQ = (2P' + 1)(2Q' + 1)$ such that P, Q, P' and Q' are all primes (i.e., P and Q are safe primes). In addition, we require that P' and Q' are two distinct primes of length $\ell(\kappa)$. For any $n \geq 1$, the n-Decisional Composite Residuosity assumption (n-DCR for short) holds iff*

$$\mathsf{Adv}^{n\text{-dcr}}_{N,\mathcal{D}}(\kappa) = \left|\Pr[\mathcal{D}(N, x^{N^n}) = 1] - \Pr[\mathcal{D}(N, x^{N^n}(1 + N)^a) = 1]\right|$$

is negligible in κ for any PPT adversary $\mathcal{D}$, where $N \leftarrow \mathcal{IG}(1^\kappa)$, $x \leftarrow \mathbb{Z}^*_{N^{n+1}}$ and $a \leftarrow \mathbb{Z}_{N^n}$.

In fact, the DCR assumption is assumed to hold for all RSA modulus. Moreover, Damgård and Jurik [10] have shown that all n-DCR assumptions are equivalent for $n \in \mathbb{N}$.

Besides the DCR assumption, we also need the following assumption.

Definition 8. *[18] For any PPT adversary $\mathcal{A}$,*

$$\mathsf{Adv}^{\text{noninv}}_{N,\mathcal{A}}(\kappa) = \Pr[x \in \mathbb{Z}^*_{N^2}, 1 < \gcd(\mathrm{ord}(x), N) < N \ : \ x \leftarrow \mathcal{A}(1^\kappa, N)]$$

is negligible in κ, where $N \leftarrow \mathcal{IG}(1^\kappa)$.

In the Paillier cryptosystem [29], each ciphertext of a plaintext $a \in \mathbb{Z}_N$ has the form $x = h^N(1 + N)^a \bmod N^2$, where $h \leftarrow \mathbb{Z}^*_{N^2}$. Since $1 < \gcd(\mathrm{ord}(x), N) < N$ implies $1 < \gcd(a, N) < N$, the above assumption actually stipulates that it is infeasible to generate Paillier encryptions of "funny messages". Note that knowing such message allows to factor N.

Let $N \leftarrow \mathcal{IG}(1^\kappa)$ and $\widetilde{N} \leftarrow \mathcal{IG}(1^\kappa)$, where $N = PQ = (2P' + 1)(2Q' + 1)$ and $\widetilde{N} = \widetilde{P}\widetilde{Q} = (2\widetilde{P}' + 1)(2\widetilde{Q}' + 1)$. Suppose the message space is $\{0,1\}^m$. Let $\mathfrak{n}$ be a positive integer such that $\mathfrak{n}(\log N - 1) \geq \log \widetilde{N} + \lambda + m + \omega(\log \kappa)$. Set $n = \lceil \mathfrak{n} \log N / \log \widetilde{N} \rceil$. Let $(ek_{\text{CH}}, td_{\text{CH}}) \leftarrow \mathsf{CH.Gen}(1^\kappa)$ be a chameleon hash function with image set $\{0,1\}^{|\widetilde{N}|/4}$. Let $\mathsf{Ext} : \mathbb{Z}^{\mathfrak{n}}_N \times \{0,1\}^d \to \{0,1\}^m$ be an average-case $(\mathfrak{n}(\log N - 1) - \log \widetilde{N} - \lambda, \epsilon_2)$-strong extractor, where ϵ_2 is a negligible function in κ. Each element $y \in \mathbb{Z}^*_{N^2}$ can be uniquely represented as $y = a + bN \bmod N^2$ $(0 \leq a, b \leq N - 1)$, and $\gcd(a, N) = 1$ must hold. Now define a map $\chi(y) = b \in \mathbb{Z}_N$. For any fixed y, if c ranges over $\{0, \ldots, N - 1\}$, then $\chi(y(1 + N)^c)$ ranges over $\mathbb{Z}_N$ as well, as shown in [9]. Applying the transformation from Section 3 to the DCR-based HPS (in Appendix C) and OT-LF (in Appendix D), we obtain a PKE scheme which is presented in Fig. 2.

Key Generation. $\mathsf{PKE}_2.\mathsf{Gen}(1^\kappa)$: Compute $g = -h^{2N} \bmod N^2$ with $h \leftarrow \mathbb{Z}_{N^2}^*$. Choose $x_1, \ldots, x_{\mathfrak{n}} \leftarrow \{0, \ldots, \lfloor N^2/2 \rfloor\}$ and compute $pk_i = g^{x_i} \bmod N^2$. Choose a random pair $(t_a^*, t_c^*) \leftarrow \{0,1\}^* \times \mathcal{R}_{\text{CH}}$ and compute $b^* = \mathsf{CH.Eval}(ek_{\text{CH}}, t_a^*; t_c^*)$. Compute $E = \widetilde{g}^{\widetilde{N}^n}(1+\widetilde{N})^{-b^*} \bmod \widetilde{N}^{n+1}$ with a random $\widetilde{g} \leftarrow \mathbb{Z}_{\widetilde{N}^{n+1}}^*$. Return $PK = (N, \mathfrak{n}, pk_1, \ldots, pk_{\mathfrak{n}}, g, \widetilde{N}, n, E, ek_{\text{CH}})$ and $SK = (x_1, \ldots, x_{\mathfrak{n}})$.

Encryption. $\mathsf{PKE}_2.\mathsf{Enc}(PK, M)$: For a public key PK and a message $M \in \{0,1\}^m$, choose a random $r \in \{0, \ldots, \lfloor N/2 \rfloor\}$ and a random seed $s \in \{0,1\}^d$. It then computes

$$C = g^r \bmod N^2,\ K = (\chi(pk_1^r), \ldots, \chi(pk_{\mathfrak{n}}^r)),\ \Psi = \mathsf{Ext}(K, s) \oplus M,\ \Pi = (E(1+\widetilde{N})^b)^K \bmod \widetilde{N}^{n+1},$$

where $b = \mathsf{CH.Eval}(ek_{\text{CH}}, t_a; t_c)$ for the auxiliary tag $t_a = (C, s, \Psi)$ and a random filter core tag $t_c \in \mathcal{R}_{\text{CH}}$. Return $CT = (C, s, \Psi, \Pi, t_c)$. Note that in the computation of Π, K is considered as an element in $\mathbb{Z}_{\widetilde{N}^n}$.

Decryption. $\mathsf{PKE}_2.\mathsf{Dec}(SK, CT)$, given a ciphertext $CT = (C, s, \Psi, \Pi, t_c)$, computes $K' = (\chi(pk_1^{x_1}), \ldots, \chi(pk_{\mathfrak{n}}^{x_{\mathfrak{n}}}))$ and $\Pi' = (E(1+\widetilde{N})^b)^{K'} \bmod \widetilde{N}^{n+1}$, where $b = \mathsf{CH.Eval}(ek_{\text{CH}}, (C, s, \Psi); t_c)$. It checks whether $\Pi = \Pi'$. If not, it rejects with $\perp$. Else, it returns $M = \Psi \oplus \mathsf{Ext}(K', s)$.

Figure 2: A DCR-based PKE Scheme $\mathsf{PKE}_2 = (\mathsf{PKE}_2.\mathsf{Gen}, \mathsf{PKE}_2.\mathsf{Enc}, \mathsf{PKE}_2.\mathsf{Dec})$

Theorem 5. *If the DCR assumption holds in $\mathbb{Z}_{N^2}^*$ and $\mathbb{Z}_{\widetilde{N}^{n+1}}^*$, and CH is a chameleon hash function, then PKE_2 is λ-LR-CCA secure for any $\lambda \le \mathfrak{n}(\log N - 1) - \log \widetilde{N} - m - \omega(\log \kappa)$. In particular, the leakage rate in PKE_2 can approach $1/2$ and*

$$\begin{aligned}\mathsf{Adv}_{\mathsf{PKE}_2,\mathcal{A}}^{\text{lr-cca}}(\kappa) \le \quad & \mathsf{Adv}_{N,\mathcal{B}_1}^{\text{dcr}}(\kappa) + 2\mathsf{Adv}_{\widetilde{N},\mathcal{B}_2}^{\text{n-dcr}}(\kappa) + Q(\kappa)\left(2\mathsf{Adv}_{\widetilde{N},\mathcal{B}_2}^{\text{n-dcr}}(\kappa) + \mathsf{Adv}_{\mathsf{CH},\mathcal{B}_3}^{\text{cr}}(\kappa)\right) \\ & + Q(\kappa)\mathsf{Adv}_{N,\mathcal{B}_4}^{\text{noninv}}(\kappa) + \frac{Q(\kappa)\widetilde{N}2^{\lambda+m}}{(N/2)^n - Q(\kappa)} + \epsilon_2\end{aligned}$$

where $Q(\kappa)$ is the number of decryption queries made by $\mathcal{A}$.

Proof. The proof of Theorem 5 is given in Appendix E. □

6 Conclusion and Further Work

We present a new generic construction of a public-key encryption scheme secure against leakage-resilient chosen-ciphertext attacks, from any ϵ-*universal* HPS and any one-time lossy filter (OT-LF). Instantiations from the DDH and DCR assumptions show that our construction is practical and achieves leakage rate of $1/2 - o(1)$. When a slightly weaker universality property of HPS holds with overwhelming probability over the choice of C from the invalid set, LR-CPA security with leakage rate of $1 - o(1)$ can be easily constructed from HPS [27]. In our construction, the HPS is required to be ϵ-*universal* for the worst-case choice of C from the invalid set $\mathcal{C} \setminus \mathcal{V}$. That is the reason why those LR-CPA security with leakage rate of $1 - o(1)$ from some HPS cannot be converted into LR-CCA security with OT-LF. The open question is how to further improve leakage rate while keeping the practicality of PKE.

Acknowledgements.

The authors thank the Asiacrypt 2013 anonymous referees for very useful comments and suggestions. This work was supported by the National Natural Science Foundation of China (Grant No. 61170229, 61373153), the Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20110073110016), and the Scientific innovation projects of Shanghai Education Committee (Grant No. 12ZZ021).

References

[1] Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous hardcore bits and cryptography against memory attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer (2009) 1

[2] Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S., Wichs, D.: Public-key encryption in the bounded-retrieval model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 113–134. Springer (2010) 1

[3] Brakerski, Z., Goldwasser, S.: Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 1–20. Springer (2010) 1

[4] Brakerski, Z., Kalai, Y.T., Katz, J., Vaikuntanathan, V.: Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In: FOCS 2010. pp. 501–510. IEEE Computer Society (2010) 1

[5] Camenisch, J., Chandran, N., Shoup, V.: A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 351–368. Springer (2009) 2

[6] Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer (2004) 2, 13

[7] Chow, S.S.M., Dodis, Y., Rouselakis, Y., Waters, B.: Practical leakage-resilient identity-based encryption from simple assumptions. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM CCS 2010. pp. 152–161. ACM (2010) 1

[8] Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer (1998) 2

[9] Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer (2002) 2, 4, 10, 11, 14, 22

[10] Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 119–136. Springer (2001) 13, 14

[11] Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Efficient public-key cryptography in the presence of key leakage. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 613–631. Springer (2010) 2, 12, 13, 14

[12] Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: Mitzenmacher, M. (ed.) STOC 2009. pp. 621–630. ACM (2009) 1

[13] Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. 38(1), 97–139 (2008) 3, 4, 13

[14] Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: FOCS 2008. pp. 293–302. IEEE Computer Society (2008) 1

[15] Faust, S., Kiltz, E., Pietrzak, K., Rothblum, G.N.: Leakage-resilient signatures. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 343–360. Springer (2010) 1

[16] Galindo, D., Herranz, J., Villar, J.L.: Identity-based encryption with master key-dependent message security and leakage-resilience. In: Foresti, S., Yung, M., Martinelli, F. (eds.) ESORICS 2012. LNCS, vol. 7459, pp. 627–642. Springer (2012) 1, 2, 12, 13, 14

[17] Halderman, J.A., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., Felten, E.W.: Lest we remember: Cold boot attacks on encryption keys. In: van Oorschot, P.C. (ed.) USENIX Security Symposium. pp. 45–60. USENIX Association (2008) 1

[18] Hofheinz, D.: All-but-many lossy trapdoor functions. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 209–227. Springer (2012) 6, 14

[19] Hofheinz, D.: Circular chosen-ciphertext security with compact ciphertexts. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 520–536. Springer (2013) 2, 5, 6

[20] Katz, J., Vaikuntanathan, V.: Signature schemes with bounded leakage resilience. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 703–720. Springer (2009) 1

[21] Kiltz, E., Pietrzak, K., Stam, M., Yung, M.: A new randomness extraction paradigm for hybrid encryption. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 590–609. Springer (2009) 4, 11

[22] Krawczyk, H., Rabin, T.: Chameleon signatures. In: NDSS 2000. The Internet Society (2000) 6

[23] Lai, J., Deng, R.H., Liu, S.: Chameleon all-but-one TDFs and their application to chosen-ciphertext security. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571. pp. 228–245. Springer (2011) 2

[24] Lewko, A.B., Rouselakis, Y., Waters, B.: Achieving leakage resilience through dual system encryption. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 70–88. Springer (2011) 1, 13

[25] Li, S., Zhang, F., Sun, Y., Shen, L.: A new variant of the Cramer-Shoup leakage-resilient public key encryption. In: Xhafa, F., Barolli, L., Pop, F., Chen, X., Cristea, V. (eds.) INCoS 2012. pp. 342–346. IEEE (2012) 2, 12, 13, 14

[26] Liu, S., Weng, J., Zhao, Y.: Efficient public key cryptosystem resilient to key leakage chosen ciphertext attacks. In: Dawson, E. (ed.) CT-RSA 2013. LNCS, vol. 7779, pp. 84–100. Springer (2013) 2

[27] Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 18–35. Springer (2009) 1, 2, 4, 15

[28] Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. SIAM J. Comput. 41(4), 772–814 (2012) 4, 12, 13, 14

[29] Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer (1999) 13, 14

[30] Pietrzak, K.: A leakage-resilient mode of operation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 462–482. Springer (2009) 1

[31] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Dwork, C. (ed.) STOC 2008. pp. 187–196. ACM (2008) 2

[32] Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer (1991) 4

[33] Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332 (2004), `http://eprint.iacr.org/` 7, 8

A Proof of Theorem 2

Sketch. To show that HPS_1 is perfectly universal, it suffices to show that for all $C = (u_1, u_2) \in \mathcal{C} \setminus \mathcal{V}$, $\Lambda_{sk}(C)$ is uniformly distributed over $\mathcal{K}$ conditioned on any fixed public key pk. Let $\Lambda_i(C) = u_1^{x_{i,1}} u_2^{x_{i,2}}$, then $\Lambda_{sk}(C) = (\mathsf{Inj}(\Lambda_1(C)), \dots, \mathsf{Inj}(\Lambda_{\mathfrak{n}}(C)))$. Given $pk = (pk_1, \dots, pk_{\mathfrak{n}})$ and $\Lambda_{sk}(C)$, we have

$$\begin{pmatrix} \log_{g_1} pk_1 \\ \vdots \\ \log_{g_1} pk_{\mathfrak{n}} \\ \log_{g_1} \Lambda_1(C) \\ \vdots \\ \log_{g_1} \Lambda_{\mathfrak{n}}(C) \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \alpha & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & \alpha \\ r_1 & r_2\alpha & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & r_1 & r_2\alpha \end{pmatrix}}_{A} \cdot \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ \vdots \\ x_{\mathfrak{n},1} \\ x_{\mathfrak{n},2} \end{pmatrix}$$

where $\alpha = \log_{g_1} g_2$, $r_1 = \log_{g_1} u_1$ and $r_2 = \log_{g_1} u_2$. Since $r_1 \neq r_2$, we have $\det(A) = \alpha^{\mathfrak{n}}(r_2 - r_1)^{\mathfrak{n}} \neq 0$. Further more, by the fact that $(x_{i,1}, x_{i,2})_{i \in [\mathfrak{n}]}$ is uniformly distributed over $(\mathbb{Z}_q \times \mathbb{Z}_q)^{\mathfrak{n}}$, it holds that each $\Lambda_i(C)$ has a uniform distribution over G. Since $\mathsf{Inj} : G \to \mathbb{Z}_q$ is an injection, this implies that $\mathsf{Inj}(\Lambda_i(C))$ leads to a uniform distribution on $\mathbb{Z}_q$. Thus, given any specific pk and $C \in \mathcal{C} \setminus \mathcal{V}$, $\Lambda_{sk}(C)$ is uniformly distributed on $\mathcal{K} = \mathbb{Z}_q^{\mathfrak{n}}$. It follows that for all $K \in \mathbb{Z}_q^{\mathfrak{n}}$, $\Pr[\Lambda_{sk}(C)) = K \mid (pk, C)] = 1/q^{\mathfrak{n}}$. □

B Proof of Theorem 3

Sketch. To prove this theorem, it suffices to prove its lossiness, indistinguishability and evasiveness.

Lossiness. Define $(y_1, \dots, y_n) := \mathsf{LF}_1.\mathsf{Eval}(Fpk, t, X) = X \cdot (E \otimes \widetilde{g}^{b\mathbf{I}})$. Define $A = (A_{i,j}) \in \mathbb{Z}_{\widetilde{q}}^{n \times n}$ with $A_{i,j} = r_i s_j$ for $i, j \in [n]$. Then $E = \widetilde{g}^{A - b^*\mathbf{I}} \in \widetilde{G}^{n \times n}$, and

$$(y_1, \dots, y_n) = X \cdot (E \otimes \widetilde{g}^{b\mathbf{I}}) = X \cdot \widetilde{g}^{A + (b - b^*)\mathbf{I}} = \widetilde{g}^{X \cdot (A + (b - b^*)\mathbf{I})}.$$

More precisely,

$$y_j = \widetilde{g}^{r_j s_j X_j + (b - b^*) X_j} \cdot \prod_{i=1, i \neq j}^{n} \widetilde{g}^{r_i s_j X_i} = \widetilde{g}^{(b - b^*) X_j} \cdot \prod_{i=1}^{n} \widetilde{g}^{r_i s_j X_i}$$

for $j \in [n]$.

If $t \in \mathcal{T}_{inj}$, then $b \notin \{b^*, b^* - \mathbf{Tr}(A)\}$. Hence matrix $B := A + (b - b^*)\mathbf{I}$ is of full rank. Therefore, $\mathsf{LF}_1.\mathsf{Eval}(Fpk, t, X) = \widetilde{g}^{X \cdot B}$ is an injective function.

On the other hand, if $t \in \mathcal{T}_{loss}$, then $b = b^*$. Hence $B = A + (b - b^*)\mathbf{I}$ is a matrix of rank 1. Obviously, $\mathsf{LF}_1.\mathsf{Eval}(Fpk, t, X) = \widetilde{g}^{X \cdot B}$ has only $\widetilde{q}$ distinct values. So, $\ell_{\mathsf{LF}_1} = \log \widetilde{q}$.

Indistinguishability. Firstly, we define the following problem. Given a group $\langle \widetilde{G}, \widetilde{g}, \widetilde{q} \rangle$, let $E^{(0)} = (\widetilde{g}^{r_i s_j})_{i,j \in [n]}$, where $r_i, s_j \leftarrow \mathbb{Z}_q$, and $E^{(1)}$ be the same as $E^{(0)}$ except that $E^{(1)}_{i,i} = \widetilde{g}^{r_i s_i} \widetilde{g}^{b'}$ for $i \in [n]$ and $b' \leftarrow \mathbb{Z}_q$. Let E' be the same as $E^{(0)}$ except that $E'_{i,i} = \widetilde{g}^{b_i}$ with $b_i \leftarrow \mathbb{Z}_q$, $i \in [n]$. We claim (with a deferred proof) that for any PPT distinguisher $\mathcal{D}$,

$$|\Pr[\mathcal{D}(\widetilde{G}, \widetilde{g}, \widetilde{q}, E^{(0)}) = 1] - \Pr[\mathcal{D}(\widetilde{G}, \widetilde{g}, \widetilde{q}, E') = 1]| \leq n\mathsf{Adv}^{\mathrm{ddh}}_{\widetilde{G}, \mathcal{B}}(\kappa), \tag{11}$$

$$|\Pr[\mathcal{D}(\widetilde{G}, \widetilde{g}, \widetilde{q}, E') = 1] - \Pr[\mathcal{D}(\widetilde{G}, \widetilde{g}, \widetilde{q}, E^{(1)}) = 1]| \leq n\mathsf{Adv}^{\mathrm{ddh}}_{\widetilde{G}, \mathcal{B}}(\kappa). \tag{12}$$

Then

$$|\Pr[\mathcal{D}(\widetilde{G}, \widetilde{g}, \widetilde{q}, E^{(0)}) = 1] - \Pr[\mathcal{D}(\widetilde{G}, \widetilde{g}, \widetilde{q}, E^{(1)})) = 1]| \leq 2n\mathsf{Adv}^{\mathrm{ddh}}_{\widetilde{G}, \mathcal{B}}(\kappa). \tag{13}$$

Next, we prove that if a PPT adversary $\mathcal{A}$ can distinguish a lossy tag from an injective one, then

$$\mathsf{Adv}^{\mathrm{ind}}_{\mathsf{LF}_1, \mathcal{A}}(\kappa) \leq |\Pr[\mathcal{D}(\widetilde{G}, \widetilde{g}, \widetilde{q}, E^{(0)}) = 1] - \Pr[\mathcal{D}(\widetilde{G}, \widetilde{g}, \widetilde{q}, E^{(1)}) = 1]|. \tag{14}$$

The theorem follows directly from Eq.(13) and (14).

To prove (14), we construct a PPT distinguisher $\mathcal{D}$, who is given $(\widetilde{G}, \widetilde{g}, \widetilde{q}, E^{(\eta)})$ and try to determine η is 0 or 1 with help of $\mathcal{A}$. $\mathcal{D}$ simulates the LF_1 environment for $\mathcal{A}$ as follows.

- Choose a chameleon hash function $\mathsf{CH} = (\mathsf{CH.Gen}, \mathsf{CH.Eval}, \mathsf{CH.Equiv})$, run $(ek_{\text{CH}}, td_{\text{CH}}) \leftarrow \mathsf{CH.Gen}(1^\kappa)$ and compute $b^* = \mathsf{CH.Eval}(ek_{\text{CH}}, t_a^*; t_c^*)$ for a random tag (t_a^*, t_c^*). Let $E := E^{(\eta)} \otimes \widetilde{g}^{-b^*\mathbf{I}}$. $\mathcal{D}$ keeps $Ftd = (td_{\text{CH}}, t_a^*; t_c^*)$ and sends $Fpk = (\widetilde{q}, \widetilde{G}, \widetilde{g}, ek_{\text{CH}}, E)$ to $\mathcal{A}$.
- $\mathcal{D}$ responds $\mathcal{A}$'s query t_a with $t_c \leftarrow \mathsf{LF.LTag}(Ftd, t_a) = \mathsf{CH.Equiv}(td_{\text{CH}}, t_a^*, t_c^*, t_a)$.
- If $\mathcal{A}$ outputs 0, which means (t_a, t_c) is a lossy tag, $\mathcal{D}$ outputs 0, indicating that $E^{(\eta)} = E^{(0)}$. Otherwise $\mathcal{D}$ outputs 1, indicating that $E^{(\eta)} = E^{(1)}$.

If $E^{(\eta)} = E^{(0)}$, then $E = E^{(0)} \otimes \widetilde{g}^{-b^*\mathbf{I}}$, and $b^* = \mathsf{CH.Eval}(ek_{\text{CH}}, t_a^*; t_c^*) = \mathsf{CH.Eval}(ek_{\text{CH}}, t_a; t_c)$. Hence (t_a, t_c) is a lossy tag. If $E^{(\eta)} = E^{(1)}$, then $E = E^{(0)} \otimes \widetilde{g}^{-(b^*-b')\mathbf{I}}$, and $\mathsf{CH.Eval}(ek_{\text{CH}}, t_a; t_c) = b^* \neq b^* - b'$, hence (t_a, t_c) is not lossy, and t_c is uniformly distributed by CH's property of equivocation. This concludes the proof of Eq.(14).

Now it remains to prove Eq.(11) and (12). For any $\mathcal{D}$ who aims to distinguish $E^{(0)}$ and $E^{(1)}$, we proceeds with a series of games.

In Game_ℓ $(\ell = 0, 1, \cdots, n)$, chooses $r_i, s_j, z_i \in \mathbb{Z}_q$ for $i, j \in [n]$. Compute $E^{(0.\ell)} = \left(E_{i,j}^{(0.\ell)}\right)_{i,j\in[n]}$ with

$$E_{i,j}^{(0.\ell)} = \begin{cases} \widetilde{g}^{r_i s_j} & i \neq j \\ \widetilde{g}^{z_i} & 1 \leq i = j \leq \ell \\ \widetilde{g}^{r_i s_i} & i = j > \ell \end{cases}.$$

Send $(\widetilde{G}, \widetilde{g}, \widetilde{q}, E^{(0,\ell)})$ to $\mathcal{D}$. Let Win_ℓ denote the event that $\mathcal{D}$ outputs 1 in Game_ℓ, then

If $\ell = 0$, then $E^{(0.\ell)} = E^{(0)}$. If $\ell = n$ then $E^{(0.\ell)} = E'$.

$$\Pr[\mathsf{Win}_0] = \Pr[\mathcal{D}(\widetilde{G}, \widetilde{g}, \widetilde{q}, E^{(0.0)}) = 1] = \Pr[\mathcal{D}(\widetilde{G}, \widetilde{g}, \widetilde{q}, E^{(0)}) = 1]; \tag{15}$$

$$\Pr[\mathsf{Win}_n] = \Pr[\mathcal{D}(\widetilde{G}, \widetilde{g}, \widetilde{q}, E^{(0.n)}) = 1] = \Pr[\mathcal{D}(\widetilde{G}, \widetilde{g}, \widetilde{q}, E') = 1] \tag{16}$$

Now we show that $\mathsf{Game}_{\ell-1}$ and Game_ℓ are indistinguishable due to the DDH assumption. Let $\mathcal{B}$ be a DDH distinguisher, who is going to determine whether $(\widetilde{g}, \widetilde{g}^x, \widetilde{g}^y, T)$ is a DDH tuple or not over group $\widetilde{G}$ of order $\widetilde{q}$. $\mathcal{B}$ simulates an environment for $\mathcal{D}$. $\mathcal{B}$ chooses $r_i, s_j, z_i \in \mathbb{Z}_q$ for $i, j \in [n], i, j \neq \ell$, and computes $\hat{E} = \left(\hat{E}_{i,j}\right)_{i,j\in[n]}$ with

$$\hat{E}_{i,j} = \begin{cases} \widetilde{g}^{r_i s_j} & i \neq \ell, j \neq \ell, i \neq j \\ (\widetilde{g}^x)^{s_j} & i = \ell, j \neq \ell \\ (\widetilde{g}^y)^{r_i} & i \neq l, j = \ell \\ \widetilde{g}^{z_i} & 1 \leq i = j < \ell \\ T & i = j = \ell \\ \widetilde{g}^{r_i s_i} & i = j > \ell \end{cases}.$$

If $(\widetilde{g}, \widetilde{g}^x, \widetilde{g}^y, T)$ is a DDH tuple, then $T = g^{xy}$ and $\hat{E} = E^{(0.\ell-1)}$, otherwise $T = g^{z_\ell}$ for some random element z_ℓ and $\hat{E} = E^{(0.\ell)}$. Consequently

$$|\Pr[\mathsf{Win}_{\ell-1}] - \Pr[\mathsf{Win}_\ell]| = |\Pr[\mathcal{D}(\widetilde{G}, \widetilde{g}, \widetilde{q}, E^{(0.\ell-1)}) = 1] - \Pr[\mathcal{D}(\widetilde{G}, \widetilde{g}, \widetilde{q}, E^{(0.\ell)}) = 1] \leq \mathsf{Adv}_{\widetilde{G},\mathcal{B}}^{\text{ddh}}(\kappa).$$

A hybrid argument gives

$$|\Pr[\mathsf{Win}_0] - \Pr[\mathsf{Win}_n] \leq n\mathsf{Adv}_{\widetilde{G},\mathcal{B}}^{\text{ddh}}(\kappa),$$

and Eq.(11) follows.

Eq.(12) can be proved in a similar way.

Evasiveness. The proof goes with two indistinguishable games. Given a PPT adversary $\mathcal{A}$ attacking on LF_1's evasiveness, let noninj_i denote the event that the output (t'_a, t'_c) by $\mathcal{A}$ is a fresh non-injective tag in Game_i. Let coll_i denote the event that $\mathcal{A}$'s output (t'_a, t'_c) induces a CH-collision in the sense that

$$\mathsf{CH.Eval}(ek_{\text{CH}}, t_a; t_c) = \mathsf{CH.Eval}(ek_{\text{CH}}, t'_a; t'_c).$$

Game_1**:** It is the original evasiveness game between adversary $\mathcal{A}$ and a challenger. The challenger calls $\mathsf{LF}_1.\mathsf{Gen}(1^\kappa)$ to obtain $Fpk = (\widetilde{q}, \widetilde{G}, \widetilde{g}, ek_{\text{CH}}, E)$ and $Ftd = (td_{\text{CH}}, t^*_a, t^*_c)$.

- The challenger sends Fpk to $\mathcal{A}$;
- $\mathcal{A}$ queries with t_a (only once), and the challenger answers with

$$t_c = \mathsf{LF}_1.\mathsf{LTag}(Ftd, t_a) = \mathsf{CH.Equiv}(td_{\text{CH}}, t^*_a, t^*_c, t_a);$$

- $\mathcal{A}$ outputs (t'_a, t'_c).

If (t'_a, t'_c) is a non-injective tag, then $\mathsf{CH.Eval}(ek_{\text{CH}}, t'_a; t'_c)$ is either b^* or $b^* - \mathbf{Tr}(A)$. Hence

$$\mathsf{Adv}^{\text{eva}}_{\mathsf{LF}_1,\mathcal{A}}(\kappa) = \Pr[\mathsf{noninj}_1] \leq \Pr[\mathsf{coll}_1] + \Pr[\mathsf{CH.Eval}(ek_{\text{CH}}, t'_a; t'_c) = b^* - \mathbf{Tr}(A)]. \tag{17}$$

Next, we show that any PPT adversary $\mathcal{A}$ that produces $\mathsf{CH.Eval}(ek_{\text{CH}}, t'_a; t'_c) = b^* - \mathbf{Tr}(A)$ can be used to compute discrete logarithms in $\widetilde{G}$ of order $\widetilde{q}$, hence contradicting the DDH assumption. Given a discrete logarithm challenge $(\widetilde{g}, \widetilde{g}^x) \in \widetilde{G}^2$, a PPT algorithm $\mathcal{B}$ is going to compute the value of x with help of $\mathcal{A}$. Now $\mathcal{B}$ simulates an environment for $\mathcal{A}$.

- Choose a chameleon hash function $\mathsf{CH} = (\mathsf{CH.Gen}, \mathsf{CH.Eval}, \mathsf{CH.Equiv})$, run $(ek_{\text{CH}}, td_{\text{CH}}) \leftarrow \mathsf{CH.Gen}(1^\kappa)$ and compute $b^* = \mathsf{CH.Eval}(ek_{\text{CH}}, t^*_a; t^*_c)$ for a random tag (t^*_a, t^*_c). Choose $r_2, \ldots, r_n, s_1, \ldots, s_n \in \mathbb{Z}_{\widetilde{q}}$, and compute $E = (E_{i,j}) \in \widetilde{G}^{n\times n}$, where

$$E_{i,j} = \begin{cases} (\widetilde{g}^x)^{s_j} & i = 1, j = 2, \ldots, n. \\ (\widetilde{g}^x)^{s_1} \cdot \widetilde{g}^{-b^*} & i = j = 1 \\ \widetilde{g}^{r_i s_j} & i \neq j, i = 2, \ldots, n, j = 1, \ldots, n \\ \widetilde{g}^{r_i s_j - b^*} & i = j, i = 2, \ldots, n \end{cases}.$$

  $\mathcal{B}$ keeps $Ftd = (td_{\text{CH}}, t^*_a; t^*_c)$ and sends $Fpk = (\widetilde{q}, \widetilde{G}, \widetilde{g}, ek_{\text{CH}}, E)$ to $\mathcal{A}$.
- $\mathcal{B}$ responds $\mathcal{A}$'s query t_a with $t_c \leftarrow \mathsf{LF.LTag}(Ftd, t_a) = \mathsf{CH.Equiv}(td_{\text{CH}}, t^*_a, t^*_c, t_a)$.
- $\mathcal{A}$ outputs (t'_a, t'_c).

$\mathcal{B}$ computes $b' = \mathsf{CH.Eval}(ek_{\text{CH}}, t'_a; t'_c)$. If $b' \neq b^*$, $\mathcal{B}$ outputs

$$x = (b^* - b' - \sum_{i=2}^{n} r_i s_i)/s_1 \bmod \widetilde{q}. \tag{18}$$

It is easy to see that $\mathcal{B}$ gives a perfect simulation for $\mathcal{A}$ by implicitly setting $r_1 = x$. Since $s_1 \leftarrow \mathbb{Z}_{\widetilde{q}}$, $s_1 = 0$ with probability $1/\widetilde{q}$. If $\mathsf{CH.Eval}(ek_{\text{CH}}, t'_a; t'_c) = b^* - \mathbf{Tr}(A)$, then $\mathbf{Tr}(A) = b^* - b' = xs_1 + \sum_{i=2}^{n} r_i s_i$ and Eq.(18) follows. Hence,

$$\Pr[\mathsf{CH.Eval}(ek_{\text{CH}}, t'_a; t'_c) = b^* - \mathbf{Tr}(A)] = \mathsf{Adv}^{\text{ddh}}_{G,\mathcal{B}}(\kappa). \tag{19}$$

Game_2**:** It is the same as Game_1, except for the response of the challenger to $\mathcal{A}$'query t_a. In this game, the challenger chooses $t_c \leftarrow \mathcal{R}_{\text{CH}}$ instead of computing it with $\mathsf{LF}_1.\mathsf{LTag}(Ftd, t_a)$.

Next, we will show any difference between the two events coll_1 and coll_2 results in a distinguisher $\mathcal{D}$ against the indistinguishability of LF_1 and

$$\mathsf{Adv}^{\mathrm{ind}}_{\mathsf{LF}_1,\mathcal{D}}(\kappa) = |\Pr[\mathsf{coll}_1] - \Pr[\mathsf{coll}_2]|. \tag{20}$$

Suppose that $\mathcal{D}$ is a distinguisher in the indistinguishability game of LF_1. $\mathcal{D}$ is given Fpk, and has one oracle access with its own choice of t_a. Given the response t_c from the oracle, $\mathcal{D}$ tells whether (t_a, t_c) is a lossy tag or not. Now $\mathcal{D}$ simulates an environment for $\mathcal{A}$.

- $\mathcal{D}$ sends Fpk to $\mathcal{A}$.
- When $\mathcal{A}$ presents a query of t_a, $\mathcal{D}$ transfers t_a to it's own oracle, and sends the oracle's answer t_c back to $\mathcal{A}$.
- When $\mathcal{A}$ outputs (t'_a, t'_c), $\mathcal{D}$ checks whether $\mathsf{CH.Eval}(ek_{\mathrm{CH}}, t_a; t_c) = \mathsf{CH.Eval}(ek_{\mathrm{CH}}, t'_a; t'_c)$. If yes, $\mathcal{D}$ outputs 1, otherwise 0.

If (t_a, t_c) is a lossy tag, $\mathcal{D}$ simulates Game_1. Otherwise, it simulates Game_2. Eq. (20) follows immediately.

Note that Ftd is not needed in Game_2. Then a straightforward reduction to the collision property of CH yields

$$\Pr[\mathsf{coll}_2] = \mathsf{Adv}^{\mathrm{cr}}_{\mathsf{CH},\mathcal{B}}(\kappa). \tag{21}$$

Combining (17) (19)(20) and (21) concludes the proof. $\square$

C A DCR-based HPS

Let $N \leftarrow \mathcal{IG}(1^\kappa)$, where $N = PQ = (2P'+1)(2Q'+1)$, P' and Q' are distinct random primes of length $\ell(\kappa)$. $P = 2P'+1$ and $Q = 2Q'+1$ are also primes. Let $N' = P'Q'$. The group $\mathbb{Z}^*_{N^2}$ can be decomposed as an internal direct product $\mathbb{Z}^*_{N^2} = G_N \times G_{N'} \times G_2 \times T$, where G_i denotes a cyclic group of order i and T the subgroup generated by $(-1 \bmod N^2)$. Define $G_{2NN'} := G_N \times G_{N'} \times T$ and $G_{2N'} := G_{N'} \times T$. Then $G_{2NN'}$ is a cyclic group of order $2NN'$ and $G_{2N'}$ is a cyclic group of order $2N'$. Choose a random $h \leftarrow \mathbb{Z}^*_{N^2}$ and set $g = -h^{2N} \bmod N^2$. It is easy to see that g is a generator of $G_{2N'}$ with overwhelming probability. Next, we describe a hash proof system $\mathsf{HPS}_2 = (\mathsf{HPS}_2.\mathsf{Gen}, \mathsf{HPS}_2.\mathsf{Pub}, \mathsf{HPS}_2.\mathsf{Priv})$.

The parameter generation algorithm $\mathsf{HPS}_2.\mathsf{Gen}(1^\kappa)$ generates a parameterized instance $\mathsf{params} = (\mathsf{group}, \mathcal{K}, \mathcal{C}, \mathcal{V}, \mathcal{SK}, \mathcal{PK}, \Lambda_{sk}, \mu)$ in the following way.

- Choose $\mathfrak{n} \in \mathbb{N}$. Set $\mathsf{group} = \langle N, g, \mathfrak{n}\rangle$, $\mathcal{C} = G_{2NN'}$ and $\mathcal{V} = G_{2N'}$ with witness $W = \{0, \ldots, \lfloor N/2 \rfloor\}$.
- Let $\mathcal{K} = \mathbb{Z}_N^{\mathfrak{n}}$, and $\mathcal{SK} = \{0, \ldots, \lfloor N^2/2 \rfloor\}^{\mathfrak{n}}$, $\mathcal{PK} = G^{\mathfrak{n}}_{2N'}$.
- For $sk = (x_i)_{i=1}^{\mathfrak{n}} \in \mathcal{SK}$, define $pk = (pk_i)_{i\in[\mathfrak{n}]} = \mu(sk) = (g^{x_i})_{i\in[\mathfrak{n}]}$.
- For all $C \in \mathcal{C}$, define $\Lambda_{sk}(C) = (\chi(C^{x_i}))_{i\in[\mathfrak{n}]}$. Recall that $\chi(y) = b \in \mathbb{Z}_N$ if $y = a + bN \in \mathbb{Z}_{N^2}$.

The public evaluation and private evaluation algorithms are defined as follows:

- For all $C \in \mathcal{V}$ with witness $r \in W$, define $\mathsf{HPS}_2.\mathsf{Pub}(pk, C, r) = (\chi(pk_1^r), \ldots, \chi(pk_{\mathfrak{n}}^r))$.
- For all $C \in \mathcal{C}$, define $\mathsf{HPS}_2.\mathsf{Priv}(sk, C) = \Lambda_{sk}(C) = (\chi(C^{x_i}))_{i=1}^{\mathfrak{n}}$.

Correctness directly follows by the definitions of μ and Λ_{sk}. The subset membership problem in HPS_2 is hard under the DCR assumption.

Next, we discuss its universal property.

Let $\mathcal{C}' = \{C : C = g^r(1+N)^a \in \mathcal{C}\backslash\mathcal{V} \wedge \gcd(a, N) \neq 1\}$. Under the Assumption 8, any PPT adversary $\mathcal{A}$, without the factorization information of N, outputs $C \in \mathcal{C}'$ with probability at most $\mathsf{Adv}^{\mathrm{noninv}}_{N,\mathcal{A}}(\kappa)$.

Thus, it suffices to discuss the universal property of HPS_2 for all $C \in (\mathcal{C} \setminus \mathcal{C}') \setminus \mathcal{V}$. According to the analysis of the DCR-based HPS by Cramer and Shoup [9], HPS_2 can be easily proved to be $\mathfrak{n}2^{-\ell(\kappa)}$ universal. Then Lemma 2 gives $\mathrm{H}_\infty\left(\Lambda_{sk}(C) \mid (pk, C)\right) \geq \ell(\kappa) - \log \mathfrak{n} - 1$. As a matter of fact, the lower bound can be improved significantly.

Theorem 6. *For all $pk = \mu(sk)$ and all $C \in (\mathcal{C} \setminus \mathcal{C}') \setminus \mathcal{V}$, $\mathrm{H}_\infty\left(\Lambda_{sk}(C) \mid (pk, C)\right) \geq \mathfrak{n}(\log N - 1)$.*

Proof. For $\mathfrak{n} = 1$, observe that if sk is sampled uniformly from the ideal key space $\mathcal{SK}^* = \{0, \ldots, 2NN' - 1\}$, then $\Lambda_{sk}(C)$ is uniformly distributed over $\mathbb{Z}_N$ for all $C \in (\mathcal{C} \setminus \mathcal{C}') \setminus \mathcal{V}$. However, elements cannot be readily chosen uniformly at random from $\mathcal{SK}^*$ without N'. Instead, we choose elements from the set $\mathcal{SK} = \{0, \ldots, \lfloor N^2/2 \rfloor\}$ randomly. Then $\Pr[sk = l] \leq \frac{(N^2/2)/(2NN')+1}{N^2/2}$ for $l \in \mathcal{SK}^*$. For all $C \in (\mathcal{C} \setminus \mathcal{C}') \setminus \mathcal{V}$ and all $pk \in G_{2N'}$, $\Pr[\Lambda_{sk}(C) = K \mid (pk, C)] \leq 2N' \cdot \frac{(N^2/2)/(2NN')+1}{N^2/2} \leq 2/N$. Hence, $\mathrm{H}_\infty(\Lambda_{sk}(C) \mid (pk, C)) \geq \log N - 1$.

For $\mathfrak{n} \geq 1$, the HPS can be viewed as $\mathfrak{n}$ independent copies of the case $\mathfrak{n} = 1$. Thus, the entropy will be $\mathfrak{n}(\log N - 1)$-bit. □

D A DCR-based OT-LF

The DCR-based OT-LF $\mathsf{LF}_2 = (\mathsf{LF}_2.\mathsf{Gen}, \mathsf{LF}_2.\mathsf{Eval}, \mathsf{LF}_2.\mathsf{LTag})$ is defined as follows.

Key Generation. $\mathsf{LF}_2.\mathsf{Gen}(1^\kappa)$: $\widetilde{N} \leftarrow \mathcal{IG}(1^\kappa)$ and $n \in \mathbb{N}$. Recall that $\widetilde{N} = (2\widetilde{P}' + 1)(2\widetilde{Q}' + 1)$ and $\widetilde{N}' = \widetilde{P}'\widetilde{Q}'$. The decomposition of $\mathbb{Z}^*_{\widetilde{N}^{n+1}}$ is given by $\mathbb{Z}^*_{\widetilde{N}^{n+1}} = G_{\widetilde{N}^n} \times G_{\widetilde{N}'} \times G_2 \times T$, where each group G_i is a cyclic group of order i and T is the subgroup of $\mathbb{Z}^*_{\widetilde{N}^{n+1}}$ generated by $(-1 \bmod \widetilde{N}^{n+1})$. Choose a random $\widetilde{g} \leftarrow \mathbb{Z}^*_{\widetilde{N}^{n+1}}$. Run $\mathsf{CH}.\mathsf{Gen}(1^\kappa)$ to obtain $(ek_{\mathrm{CH}}, td_{\mathrm{CH}})$. Pick a random pair $(t_a^*, t_c^*) \in \{0,1\}^* \times \mathcal{R}_{\mathrm{CH}}$ and compute $b^* = \mathsf{CH}.\mathsf{Eval}(ek_{\mathrm{CH}}, t_a^*; t_c^*) \in \{0,1\}^{|\widetilde{N}|/4}$. Set $E = \widetilde{g}^{\widetilde{N}^n}(1 + \widetilde{N})^{-b^*} \bmod \widetilde{N}^{n+1}$. Finally, output $Fpk = (\widetilde{N}, n, ek_{\mathrm{CH}}, E)$ and $Ftd = (td_{\mathrm{CH}}, t_a^*, t_c^*)$. The tag space is defined as $\mathcal{T} = \{0,1\}^* \times \mathcal{R}_{\mathrm{CH}}$. The lossy tag is defined as $\mathcal{T}_{loss} = \{(t_a, t_c) : (t_a, t_c) \in \mathcal{T} \wedge \mathsf{CH}.\mathsf{Eval}(ek_{\mathrm{CH}}, t_a; t_c) = b^*\}$. Particularly, $\mathcal{T}_{inj} = \mathcal{T} \setminus \mathcal{T}_{loss}$.

Evaluation. For a tag $t = (t_a, t_c) \in \{0,1\}^* \times \mathcal{R}_{\mathrm{CH}}$ and an input $X \in \mathbb{Z}_{\widetilde{N}^n}$, $\mathsf{LF}_2.\mathsf{Eval}(Fpk, t, X)$ computes

$$y = (E(1 + \widetilde{N})^b)^X \bmod \widetilde{N}^{n+1}$$

where $b = \mathsf{CH}.\mathsf{Eval}(ek_{\mathrm{CH}}, t_a; t_c)$

Clearly, if $t \in \mathcal{T}_{inj}$, then $\gcd(b - b^*, \widetilde{N}) = 1$. Hence, the order of $E(1 + \widetilde{N})^b$ is at least $\widetilde{N}^n$. So, $\mathsf{LF}_2.\mathsf{Eval}(Fpk, t, X)$ computes an injective function. On the other hand, if $t \in \mathcal{T}_{loss}$, we have

$$y = \left(\widetilde{g}^{\widetilde{N}^n}(1 + \widetilde{N})^{-b^*}(1 + \widetilde{N})^{b^*}\right)^X = \left(\widetilde{g}^{\widetilde{N}^n}\right)^X \bmod \widetilde{N}^{n+1}.$$

Obviously, $\widetilde{g}^{\widetilde{N}^n}$ has order at most $2N'$. Thus $\ell_{\mathsf{LF}_2} = \log(2\widetilde{N}') < \log \widetilde{N}$.

Lossy Tag Generation. For an auxiliary tag t_a, $\mathsf{LF}_2.\mathsf{LTag}(Ftd, t_a)$ computes

$$t_c = \mathsf{CH}.\mathsf{Equiv}(td_{\mathrm{CH}}, t_a^*, t_c^*, t_a).$$

Theorem 7. LF_2 *is an $(\mathbb{Z}_{\widetilde{N}^n}, \log \widetilde{N})$-OT-LF under the DCR assumption.*

Proof. As we have discussed the lossiness, it remains to show the indistinguishability and evasiveness.

Indistinguishability. For a fixed PPT adversary $\mathcal{A}$, we proceed through a sequence of games. In each game, $\mathcal{A}$ interacts with an oracle once to get (t_a, t_c), then outputs 1 (indicating that (t_a, t_c) is lossy) or 0 (indicating (t_a, t_c) is injective). Let out_i denote $\mathcal{A}$'s output in Game_i.

Game_0: The challenger generates $Fpk = (N, n, ek_{\text{CH}}, E)$ and $Ftd = (td_{\text{CH}}, t_a^*, t_c^*)$ with $\mathsf{LF}_2.\mathsf{Gen}(1^\kappa)$, and sends Fpk to $\mathcal{A}$. Then $\mathcal{A}$ issues a query of t_a, and the challenger answers with $t_c^{(0)} \leftarrow \mathsf{LF}_2.\mathsf{LTag}(Ftd, t_a) = \mathsf{CH.Equiv}(td_{\text{CH}}, t_a^*, t_c^*, t_a)$. Finally, $\mathcal{A}$ outputs 1 or 0.

Then

$$\Pr[\mathsf{out}_0 = 1] = \Pr[\mathcal{A}(Fpk, (t_a, t_c^{(0)})) = 1].$$

Game_1: It proceeds just like Game_0 except for the generation of E in Fpk. In this game, $Fpk = (\widetilde{N}, n, ek_{\text{CH}}, E')$ and $E' = \widetilde{h}(1+\widetilde{N})^{-b^*} \bmod \widetilde{N}^{n+1}$, where $\widetilde{h} \leftarrow \mathbb{Z}^*_{\widetilde{N}^{n+1}}$. A straightforward reduction to the n-DCR assumption yields that

$$|\Pr[\mathsf{out}_1 = 1] - \Pr[\mathsf{out}_0 = 1]| = \mathsf{Adv}^{n\text{-dcr}}_{\widetilde{N},\mathcal{D}}(\kappa)$$

for a n-DCR distinguisher $\mathcal{D}$.

Game_2: It is the same as Game_1, except the way how the challenger answers $\mathcal{A}$'s query t_a. The challenger chooses $t_c^{(1)}$ from $\mathcal{R}_{ch}$ uniformly at random. Since $E = \widetilde{h}(1+\widetilde{N})^{-b^*}$ for a random $\widetilde{h} \leftarrow \mathbb{Z}^*_{\widetilde{N}^{n+1}}$, the part $\widetilde{h}$ completely hides the information of b^*. So, Game_2 and Game_1 are the same. Hence $\Pr[\mathsf{out}_2 = 1] = \Pr[\mathsf{out}_1 = 1]$.

Game_3: It is the same as Game_2, except that the component E' in Fpk is replaced by E, which is generated just like in Game_0, i.e., $E = \widetilde{g}^{\widetilde{N}^n}(1+\widetilde{N})^{-b^*}$. Another straightforward reduction to the n-DCR assumption yields that

$$|\Pr[\mathsf{out}_3 = 1] - \Pr[\mathsf{out}_2 = 1]| = \mathsf{Adv}^{n\text{-dcr}}_{\widetilde{N},\mathcal{D}'}(\kappa),$$

for a suitable n-DCR distinguisher $\mathcal{D}'$.

Observe that in Game_3, the public key Fpk is the same as in $(Fpk, Ftd) \leftarrow \mathsf{LF}_2.\mathsf{Gen}(1^\kappa)$ and $t_c^{(1)} \leftarrow \mathcal{R}_{ch}$.

$$\Pr[\mathsf{out}_3 = 1] = \Pr[\mathcal{A}(Fpk, (t_a, t_c^{(1)})) = 1].$$

By definition of indistinguishability for LF, we have

$$\mathsf{Adv}^{\text{ind}}_{\mathsf{LF},\mathcal{A}}(\kappa) : \quad = \quad |\Pr[\mathsf{out}_0 = 1] - \Pr[\mathsf{out}_3 = 1]| \leq 2\mathsf{Adv}^{n\text{-dcr}}_{\widetilde{N},\mathcal{D}'}(\kappa)$$

Evasiveness. The evasiveness comes from the indistinguishability property of LF_2 and the collision resistance of the chameleon hash function, similar to the evasiveness proof of LF_1.

$$\mathsf{Adv}^{\text{eva}}_{\mathsf{LF}_2,\mathcal{A}}(\kappa) := \mathbf{Adv}^{\text{ind}}_{\mathsf{LF}_2,\mathcal{B}}(\kappa) + \mathsf{Adv}^{\text{cr}}_{\mathsf{CH},\mathcal{D}}(\kappa).$$

We omit it here. □

E Proof of Theorem 5

The security proof of PKE_2 does not exactly follow the general proof of Theorem 1 due to the existence of "*bad* " ciphertexts constituting $\mathcal{C}'$. Nevertheless, we can craft the proof to fit Theorem 1 by adding a new game, say Game_4', between Game_4 and Game_5. Game_4' aims to deal with those bad ciphertexts submitted by the adversary during decryption queries. In this game, we just add a special rule to reject those ciphertexts in $\mathcal{C}'$. Under the Assumption 8, Game_4 and Game_4' are indistinguishable except with probability at most $Q(\kappa)\mathsf{Adv}^{\text{noninv}}_{N,\mathcal{A}}(\kappa)$ even if the adversary knows the HPS secret key. Afterwards, we can directly use the universality of HPS_2 on all $C \in (\mathcal{C} \setminus \mathcal{C}') \setminus \mathcal{V}$ in the following games, exactly like the proof in Theorem 1.