SCIENCE CHINA
Information Sciences

• **Supplementary File** •

LR-RRA-CCA Secure Functional Encryption for Randomized Functionalities from Trapdoor HPS and LAF

Huige WANG[1,2], Kefei CHEN[3*], Baodong QIN[4,5] & Ziyuan HU[1]

[1] *Department of Network Engineering, Anhui Science and Technology University, Fengyang* 233100*, China;*
[2] *Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai* 200240*, China;*
[3] *Department of Mathematics, Hangzhou Normal University, Hangzhou* 311121*, China;*
[4] *National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an* 710121*, China ;*
[5] *State Key Laboratory of Cryptology, P.O.Box5159, Beijing* 100878*, China*

Appendix A Introduction

In appendixes below, we give some supplementary materials for our short paper. In Appendix B, we give some notations; in Appendix C, we give the definition of function restrictions; in Appendix D, we give the notion of the correlated-input secure hash functions; in Appendix E, we give some related notions with trapdoor hash proof system; in Appendix F, we give some properties of Lossy Algebraic Filter (LAF); in Appendix G, we give the fomal definition of public-key functional encryption (PK-FE); in Appendix H, we give the security proof of the concrete construction for PK-FE.

Appendix B Notations

Throughout the paper, $\mathbb{N}$ denotes the set of natural numbers and $\lambda \in \mathbb{N}$ denotes the security parameter. Let $y \leftarrow A(x_1, \cdots; R)$ denote the operation of running algorithm A on inputs $x_1, \cdots$ and coins R to output y. For simplicity, we write $y \leftarrow A(x_1, \cdots; R)$ as $y \leftarrow_\$ A(x_1, \cdots)$ with implied coins. If $n \in \mathbb{N}$, we let $[n]$ denote the set $\{1, \cdots, n\}$. We call a function *negl* negligible in λ if $negl(\lambda) \in \lambda^{-\omega(1)}$ and a function *poly* a polynomial if $poly \in \lambda^{\mathcal{O}(1)}$. If X is a random variable over the set S, then we write $\max_{a \in S} \Pr[X = a]$ to denote the predictability of X and $-\log(\max_{a \in S} \Pr[X = a])$ denote the min-entropy $\mathsf{H}_\infty(X)$ of X. If $\overrightarrow{x}$ denotes a vector, then $|\overrightarrow{x}|$ denotes the number of components in $\overrightarrow{x}$. If P denotes circuit, then we use notation $\mathsf{P}_{[z]}(\cdot)$ to emphasize the fact that the value z is hard-coded into P. $x \leftarrow_\$ Sample_{\mathcal{D}}(r)$ denotes an efficiently computable sampler *Sample* which on input a randomness $r \leftarrow_\$ \mathcal{R}$, outputs a uniform and random x sampled from $\mathcal{D}$, where $\mathcal{D}$ denotes an efficiently sampleable domain and $\mathcal{R}$ denotes the random number space.

In the following, for security definition and proofs we use a code-based game playing framework in [1,7]. A game G has a main procedure, and possibly other procedure. G begins by executing the main procedure which runs an adversary A after some initialization. A can make oracle calls permitted by G. When A finishes execution, G continues to execute with A's output. By $\mathsf{G}^A \Rightarrow y$, we denote the event that G executes with A to output y. Generally, we abbreviate $\mathsf{G}^A \Rightarrow true$ or $\mathsf{G}^A \Rightarrow 1$ as G, and boolean flags and sets are initialized to false and $\emptyset$ respectively.

Appendix C Function Restrictions

In this section, we give some restrictions on the collection of functions $\Phi = \{\Phi_\lambda\}_{\lambda \in \mathbb{N}}$ that the adversary is allowed to access in its queries. These restrictions are necessary to preventing trivial attacks in our schemes. Note that the functions here refer to those the adversary chooses to act on the random numbers in our security notions.

Definition 1 (Output-Unpredictability for Functions Φ_λ)**.** . For all sufficiently large $\lambda \in \mathbb{N}$, let Φ_λ be a set of functions from $\mathcal{R}_\lambda$ to $\mathcal{R}_\lambda$ and $\alpha = poly_1(\lambda)$, $\beta = poly_2(\lambda)$ be positive integers. Then Φ_λ is (α, β)-output-unpredictability, if the probability defined below

$$\max_{P \subseteq \Phi_\lambda, X \subseteq \mathcal{R}_\lambda, |P| \leqslant \alpha, |X| \leqslant \beta} \{\mathsf{Pr}[r \leftarrow_\$ \mathcal{R}_\lambda : \{\phi(r) : \phi \in P\} \cap X \neq \emptyset]\},$$

* Corresponding author (email: kfchen@hznu.edu.cn)

is negligible in λ.

Definition 2 (Collision-Resistance for Functions Φ_λ)**.** . For all sufficiently large $\lambda \in \mathbb{N}$, let Φ_λ be a set of functions from $\mathcal{R}_\lambda$ to $\mathcal{R}_\lambda$ and $\alpha = poly(\lambda)$ be positive integers. Then Φ_λ is α-collision-resistance, if the probability defined below

$$\max_{P \subseteq \Phi_\lambda, |P| \leqslant \alpha} \{\Pr[r \leftarrow_\$ \mathcal{R}_\lambda : |\{\phi(r) : \phi \in P\}| \leqslant |P|]\},$$

is negligible in λ.

Let Φ be a family of functions with output-unpredictability and collision-resistance, we call $\mathcal{A}$ Φ-restricted adversary if the functions that the adversary chooses to act on the randomness belongs to Φ. In addition, note that, throughout this paper, we assume that Φ implicitly excludes all constant functions.

Appendix D Correlated-Input Secure (CIS) Hash Functions

The security notion of multi-key selective correlated-input pseudorandomness (MK-SCI-PR) [5] for the family of keyed hash functions $\mathcal{H}$ via the game shown in Figure D1.

$\underline{\mathsf{Initialise}(\lambda, l)}$
$desc = (\mathsf{CIH}.\mathcal{K}, \mathsf{CIH}.\mathcal{D}, \mathsf{CIH}.\mathcal{R}, \mathsf{h}) \leftarrow \mathsf{GenFun}(1^\lambda)$;
For $i = 1$ to l
  $k_i \xleftarrow{\$} \mathsf{CIH}.\mathcal{K}$;
$x \xleftarrow{\$} \mathsf{CIH}.\mathcal{D}$;
$b \xleftarrow{\$} \{0, 1\}$;
$\mathcal{S}' \leftarrow \emptyset$;
$fq \leftarrow false$, $ch \leftarrow false$;
$b' \leftarrow \mathcal{A}^{\mathsf{Hash},\mathsf{Func},\mathsf{Chal}}(1^\lambda, desc)$.
 	
$\underline{\mathsf{Hash}(i, j)}$
If $fq = false$ or $ch = true$
  then return $\perp$;
$\mathcal{S}' \leftarrow \mathcal{S}' \cup \{(i, j)\}$;
Return $\mathsf{h}_{k_i}(\phi_j(x))$.
	$\underline{\mathsf{Func}(\phi_1, \cdots, \phi_{q_\phi})}$
	If $fq = true$, return $\perp$;
	$fq \leftarrow true$;
	Return $k_1, \cdots, k_l$.
	
	$\underline{\mathsf{Chal}(i^*, j^*)}$
	If $fq = false$ or $ch = true$
	  return $\perp$;
	If $(i^*, j^*) \in \mathcal{S}'$
	  return $\perp$;
	$y_0 \xleftarrow{\$} \mathsf{CIH}.\mathcal{R}$;
	$y_1 \leftarrow \mathsf{h}_{k_{i^*}}(\phi_{j^*}(x))$;
	$ch = true$
	Return y_b.
	
	$\underline{\mathsf{Finalise}(b')}$
	If $b = b'$ Return 1.

Figure D1 Game l-MK-SCI-PR for a family $\mathcal{H}$ of keyed hash functions defined by GenFun.

Definition 3 (l-mk-sci-pr Security for CIS Hash Function [5])**.** A family $\mathcal{H}$ of keyed hash functions is said to be l-mk-sci-pr secure if for all Φ_λ-restricted adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ against $\mathcal{H}$ defined as

$$\mathsf{Adv}^{l\text{-mk-sci-pr}}_{\mathcal{H},\mathcal{A}}(\lambda) := 2.\Pr[l\text{-MK-SCI-PR}^{\mathcal{A}}_{\mathcal{H}}(\lambda) \Rightarrow 1] - 1,$$

is negligible in the security parameter λ.

Appendix E Trapdoor Hash Proof System (THPS)

Here, we redefine the hash proof system (HPS), which we call trapdoor hash proof systems (THPS), by additionally providing some properties such as witness-invertibility on the original HPS proposed by Cramer et al. [2]. Briefly, the witness-invertibility says that under a universal trapdoor, the witness for a valid ciphertext can be effectively recovered.

Definition 4 (Trapdoor Projective Hash Function)**.** Let $\mathcal{PK}$ be a public key set, $\mathcal{SK}$ a secret key set, $\mathcal{K}$ an encapsulated key set, $\mathcal{TK}$ a trapdoor set, $\mathcal{C}$ a ciphertext set and $\mathcal{V} \in \mathcal{C}$ a valid ciphertext set and we assume that there exists efficient algorithms which can sample $sk \leftarrow_\$ \mathcal{SK}$, $(C, w) \leftarrow_\$ \mathcal{V}$ and $C \leftarrow_\$ \mathcal{C} \backslash \mathcal{V}$, where w is a witness indicating $C \in \mathcal{V}$. Let Λ_{sk} be a hash function indexed with $sk \in \mathcal{SK}$ that maps ciphertexts to encapsulated keys. The hash function Λ_{sk} is **projective** if there exists a projection function $\mu : \mathcal{SK} \to \mathcal{PK}$ such that $\mu(sk) \in \mathcal{PK}$ determines the behavior of Λ_{sk} over the subset $\mathcal{V}$ of valid ciphertexts. Moreover, we say that Λ_{sk} is **trapdoor projective hash function** if there exists a **trapdoor** $td \in \mathcal{TK}$ and a PPT algorithm Invert which on input the trapdoor td and a ciphertext $C \in \mathcal{V}$, recovers the witness $w \leftarrow \mathsf{Invert}(td, C)$ from C. Besides, we assume that both $\Lambda_{(\cdot)}$ and μ are efficiently computable.

Definition 5 (Universal [2])**.** A trapdoor projective hash function Λ_{sk} is ϵ-universal, if for all $pk \in \mathcal{PK}$, $C \in \mathcal{C} \backslash \mathcal{V}$, $K \in \mathcal{K}$ and $td \in \mathcal{TK}$, the probability $\Pr[\Lambda_{sk}(C) = K | (pk, C, td)] \leqslant \epsilon_1$ holds and it follows that $\mathsf{H}_\infty(\Lambda_{sk}(C) | (pk, C, td)) \geqslant \log(1/\epsilon_1)$, where the probability is over all $sk \in \mathcal{SK}$ with $pk = \mu(sk)$.

Definition 6 (Subset Membership Problem (SMP) [6])**.** We say that the subset membership problem with respect to a trapdoor hash proof system THPS holds if the ciphertexts $C_0 \leftarrow_\$ \mathcal{V}$ and $C_1 \leftarrow_\$ \mathcal{C} \backslash \mathcal{V}$ are computationally indistinguishable, formally, if for all PPT adversary $\mathcal{A}$, the advantage function $\mathsf{Adv}^{\mathsf{smp}}_{\mathsf{THPS},\mathcal{A}}$ defined below

$$\mathsf{Adv}^{\mathsf{smp}}_{\mathsf{THPS},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(\mathcal{C}, \mathcal{V}, C_0) = 1 | C_0 \leftarrow_\$ \mathcal{V}] - \Pr[\mathcal{A}(\mathcal{C}, \mathcal{V}, C_1) = 1 | C_1 \leftarrow_\$ \mathcal{C} \backslash \mathcal{V}]|,$$

is negligible in the security parameter λ.

In addition, note that C_0 and C_1 can be easily distinguished with the universal trapdoor td.

Appendix F Lossy Algebraic Filter (LAF) [4]

Indistinguishability. Lossy tags are indistinguishable from random ones. Formally, for all PPT adversary $\mathcal{A}$, if the advantage function $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LAF},\mathcal{A}}(\lambda)$ defined below

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LAF},\mathcal{A}}(\lambda) = \mathsf{Pr}[\mathcal{A}(1^\lambda, lpk)^{\mathsf{LAF.Ltag}(ltk,\cdot)} = 1] - \mathsf{Pr}[\mathcal{A}(1^\lambda, lpk)^{\mathcal{O}_{\mathcal{T}_c}(\cdot)} = 1],$$

is negligible in λ, where $(lpk, ltk) \leftarrow \mathsf{LAF.KG}(1^\lambda)$ and $\mathcal{O}_{\mathcal{T}_c}(\cdot)$ is the oracle that samples a random core tag t_c.

Evasiveness. Non-injective tags are hard to find, even if given multiple lossy tags. Formally, for all PPT adversary $\mathcal{A}$, the advantage function $\mathsf{Adv}^{\mathsf{evs}}_{\mathsf{LAF},\mathcal{A}}$ defined below

$$\mathsf{Adv}^{\mathsf{evs}}_{\mathsf{LAF},\mathcal{A}}(\lambda) = \mathsf{Pr}[t \in \mathcal{T}\backslash\mathcal{T}_{inj} | t \leftarrow \mathcal{A}(1^\lambda, lpk)^{\mathsf{LAF.Ltag}(ltk,\cdot)}],$$

is negligible in λ, where $(lpk, ltk) \leftarrow \mathsf{LAF.KG}(1^\lambda)$ and $t = (t_a, t_c)$ is a non-injective tag such that t_c is not obtained via oracle $\mathsf{LAF.Ltag}(ltk, \cdot)$.

Appendix G Public-Key Functional Encryption (PK-FE) for Randomized Functions

In this section, we adopt the definitions of public-key functional encryption (PK-FE) for randomized functions in [3]. Here we simply review the definition and give our proposed security notion. Likewise, let $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$, $\mathcal{R} = \{\mathcal{R}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\mathcal{Y} = \{\mathcal{Y}_\lambda\}_{\lambda\in\mathbb{N}}$ denote three finite sets. Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a finite set of randomized functions. Each function $f \in \mathcal{F}_\lambda$ takes as input a string $x \in \mathcal{X}_\lambda$ and a randomness $r \in \mathcal{R}_\lambda$ and outputs $f(x; r) \in \mathcal{Y}_\lambda$.

Appendix G.1 PK-FE for Randomized Functions [3]

We denote a public-key functional encryption for randomized function space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ by $\mathsf{rFE} = (\mathsf{rFE.Setup}, \mathsf{rFE.KG}, \mathsf{rFE.E}, \mathsf{rFE.D})$ over plaintext space $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$. All the fourth algorithms are PPT.

Setup. The algorithm $\mathsf{rFE.Setup}(1^\lambda)$ takes as input 1^λ and outputs a master public key mpk, a master secret key msk and a trapdoor td. Note that the trapdoor td is designed to recover a witness of a ciphertext $c \in \mathcal{V}$ and is only used in the security proof of the scheme.

Key Generation. The key generation algorithm $\mathsf{rFE.KG}(msk, f)$ takes as input the master secret key msk and a function $f \in \mathcal{F}_\lambda$ and outputs the secrete key sk_f.

Encryption. The encryption algorithm $\mathsf{rFE.E}(mpk, x)$ takes as input the master public key mpk and a plaintext $x \in \mathcal{X}_\lambda$ and outputs a ciphertext ct.

Decryption. The decryption algorithm $\mathsf{rFE.D}(sk_f, ct)$ takes as input a secret key sk_f and a ciphertext ct, which encrypts plaintext x, outputs $f(x)$ or $\perp$.

The correctness for the PK-FE scheme for randomized function rFE requires that for all polynomial $n'' = n''(\lambda)$, all $\overrightarrow{f} \in \mathcal{F}_\lambda^{n''}$ and all $\overrightarrow{x} \in \mathcal{X}_\lambda^{n''}$, the following two distributions are computationally indistinguishable:

1. **Real** : $\{\mathsf{rFE.D}(sk_{f_j}, ct_i)\}_{i=1,j=1}^{n'',n''}$, where:
   - $(mpk, msk) \leftarrow \mathsf{rFE.Setup}(1^\lambda)$
   - $ct_i \leftarrow \mathsf{rFE.E}(mpk, x_i)$ for $i \in [n'']$, $x_i \in \overrightarrow{x}$
   - $sk_{f_j} \leftarrow \mathsf{rFE.KG}(msk, f_j)$ for $j \in [n'']$, $f_j \in \overrightarrow{f}$
2. **Ideal** : $\{f_j(x_i; r_{i,j})\}_{i=1,j=1}^{n'',n''}$, where $r_{i,j} \leftarrow_\$ \mathcal{R}_\lambda$.

Appendix G.2 Simulation-Based LR-RRA-CCA Security for PK-FE for Randomized Functions

In this section, we define a new security model for public-key functional encryption for randomized functions, called simulation-based LR-RRA-CCA security (short for "Sim-LR-RRA-CCA security"), by combining the notions of leakage-resilient (LR), related-randomness attacks (RRA) [5] and chosen-ciphertext attacks (CCA) for public-key encryption. Clearly, our Sim-LR-RRA-CCA security is stronger than the one proposed in [3]. Let rFE=($\mathsf{rFE.Setup}$, $\mathsf{rFE.KG}$, $\mathsf{rFE.E}$, $\mathsf{rFE.D}$) denote a public key functional encryption scheme for randomized functions, our Sim-LR-RRA-CCA security notion for rFE is defined via the games in Figure G1. Here, we require $\mathcal{A}$ to be a Φ-restricted adversary where Φ is a deterministic function set, but we do not strictly restrict the type of the set Φ, which can be a set of polynomial functions or affine functions or other functions which relies on the concrete instantiations. For simplicity, applying the Lemma 1 from [5], we only consider the case that the adversary uses one randomness index.

DISCUSSION. We extend the existing definition from [3] to include the security against related-randomness attacks and master key-leakage in our security notion in which a target key pair (mpk^*, msk^*) for the FE scheme rFE is honestly provided, where mpk^* denotes the target master public key and msk^* denotes the target master secret key. In order to formalize the intuition on the security against RRA, we allow the adversary to control which random values (since we use one randomness case in our definition, the random value here refers to the unique r initialized at the beginning of the game) and functions will be used in oracle queries where the random values are needed. To obtain the LR security, we also allow the adversary to learn a bounded amount of information about the target master secret key msk^*. Like the CCA security in the standard definition of Goyal et al., the adversary is also allowed to have access to a regular decryption oracle with the private keys sk_g which is generated with msk^* in the real world; while in the ideal world, the simulator must be able to "compute" the plaintext x from each decryption query and output $f'(x;r)$ for some true random value r. The adversary is considered successful if it distinguishes the output of the boolean relation R applied to its interaction in the real world from the output of R applied to its interaction in the ideal world.

Experiment $\mathsf{LRRRACCAREAL}^{\mathcal{A}}_{\mathsf{rFE},l_L,\mathcal{F},\mathsf{R}}(\lambda)$

$ChS \leftarrow \emptyset;\ XS \leftarrow \emptyset;\ fS \leftarrow \emptyset;\ gS \leftarrow \emptyset;$
$target \leftarrow false;\ lrq \leftarrow false;\ fq \leftarrow false;$
$LRS \leftarrow \emptyset;\ ENS \leftarrow \emptyset;\ DES \leftarrow \emptyset;$
$lfS \leftarrow \emptyset;\ KGS \leftarrow \emptyset;\ ctr \leftarrow 0;$
$(\overrightarrow{x}, (\phi_1, \cdots, \phi_{q_\phi}), st_1) \leftarrow \mathcal{A}_1(1^\lambda);$
For $i = 1$ to l
  $(mpk_i, msk_i, td_i) \leftarrow \mathsf{rFE.Setup}(1^\lambda);$
$\alpha \leftarrow \mathcal{A}_2^{\mathsf{Func},\mathsf{Target},\mathsf{KeyGen},\mathsf{ENC},\mathsf{LR},\mathsf{DEC},\mathsf{LEAK}}(st_1).$

$\underline{\mathsf{Func}(\phi_1, \cdots, \phi_{q_\phi})}$
If $fq = true$ return $\perp$;
$fq = true$; Return $\{mpk_i\}_{i\in[l]}$.

$\underline{\mathsf{Target}(j)}$
If $target = true$
  then return $\perp$;
$(mpk^*, msk^*, td^*) \leftarrow (mpk_j, msk_j, td_j);$
$target \leftarrow true;$
Return $\{msk_i\}_{i\neq j}$.

$\underline{\mathsf{LEAK}^{l_L,\lambda}_{msk^*}(f'')}$
If $fq = false$ or $target = false$
  then return $\perp$;
$lfS \leftarrow lfS \cup f''$; Return $f''(msk^*)$.

$\underline{\mathsf{KeyGen}(f, 1, n)}$
If $fq = false$ or $target = false$ or $lrq = false$ or
$\phi_n \in (KGS \cup ENS \cup DES \cup LRS)$ or
$((\exists x \in \overrightarrow{x}) \cap (f(0^{|x|}; r) \neq f(x; r)))$
  then return $\perp$, where $r \in \mathcal{R}_\lambda$;
$sk_f \leftarrow \mathsf{rFE.KG}(msk^*, f; \phi_n(r));\ fS \leftarrow fS \cup \{f\};$
$KGS \leftarrow KGS \cup \{\phi_n\};$
Return sk_f.

$\underline{\mathsf{ENC}(mpk, x, 1, n)}$
If $fq = false$ or $target = false$ or $(x \in \overrightarrow{x})$ or
  $\phi_n \in (LRS \cup KGS)$ or $(mpk \notin \{mpk_i\}_{i\in[l]})$
  then return $\perp$;
$ct \leftarrow \mathsf{rFE.E}(mpk, x; \phi_n(r));\ ENS \leftarrow ENS \cup \{\phi_n\};$
Return ct.

$\underline{\mathsf{LR}(\{x_i\}_{i\in[q_{lr}]}, 1, n)}$
If $fq = false$ or $target = false$ or
  $\phi_n \in (ENS \cup DES)$
  then return $\perp$;
For each $i \in [q_{lr}]$
  compute $ct_i^* \leftarrow \mathsf{rFE.E}(mpk^*, x_i; \phi_n(r));$
  $ctr = ctr + 1;$
$LRS \leftarrow LRS \cup \{\phi_n\};\ ChS \leftarrow ChS \cup \{ct_i^*\}_{i\in[q_{lr}]};$
If $ctr = q_{lr}$ then $lrq \leftarrow true$;
Return $\{ct_i^*\}_{i\in[q_{lr}]}$.

$\underline{\mathsf{DEC}_{msk^*}(ct, g, 1, n)}$
If $fq = false$ or $target = false$ or $ct \in ChS$ or
  $\phi_n \in (LRS \cup KGS)$
  then return $\perp$;
$sk_g \leftarrow \mathsf{rFE.KG}(msk^*, g; \phi_n(r));$
$DES \leftarrow DES \cup \{\phi_n\};\ gS \leftarrow gS \cup \{g\};$
Return $y = \mathsf{rFE.D}(sk_g, ct)$.

$\underline{\mathsf{Finalise}(\alpha)}$
Return $\mathsf{R}(\overrightarrow{x}, fS, gS, \{y\}, lfS, \alpha)$.

Experiment $\mathsf{LRRRACCAIDEAL}^{\mathcal{S}}_{\mathsf{rFE},l_L,\mathcal{F},\mathsf{R}}(\lambda)$

$ChS \leftarrow \emptyset;\ XS \leftarrow \emptyset;\ fS \leftarrow \emptyset;\ gS \leftarrow \emptyset;$
$target = false;\ lrq = false;\ fq \leftarrow false;$
$KGS \leftarrow \emptyset;\ LRS \leftarrow \emptyset;\ ENS \leftarrow \emptyset;$
$lfS \leftarrow \emptyset;\ DES \leftarrow \emptyset;\ ctr \leftarrow 0;$
$(\overrightarrow{x}, (\phi_1, \cdots, \phi_{q_\phi}), st_1) \leftarrow \mathcal{A}_1(1^\lambda);$
$(\{mpk_i\}_{i\in[l]}, st') \leftarrow \mathsf{S}_1(1^\lambda);$
$\alpha \leftarrow \mathcal{A}_2^{\mathsf{Func}',\mathsf{Target}',\mathsf{KeyGen}',\mathsf{ENC}',\mathsf{LR}',\mathsf{DEC}',\mathsf{LEAK}'}(st_1).$

$\underline{\mathsf{Func}'(\phi_1, \cdots, \phi_{q_\phi})}$
If $fq = true$ return $\perp$;
$fq = true$; Return $\{mpk_i\}_{i\in[l]}$.

$\underline{\mathsf{Target}'(j)}$
If $target = true$
  then return $\perp$;
$(mpk^*, msk^*, td^*) \leftarrow (mpk_j, msk_j, td_j);$
$target \leftarrow true;$
Return $\{msk_i\}_{i\neq j}$.

$\underline{\mathsf{LEAK}'^{l_L,\lambda}_{msk^*}(f'')}$
If $fq = false$ or $target = false$
  then return $\perp$;
$lfS \leftarrow lfS \cup f''$; Return $f''(msk^*)$.

$\underline{\mathsf{KeyGen}'(f', 1, n)}$
If $fq = false$ or $target = false$ or $lrq = false$ or
  $\phi_n \in (KGS \cup ENS \cup DES \cup LRS)$
$((\exists x \in \overrightarrow{x}) \cap (f'(0^{|x|}; r) \neq f'(x; r)))$
  then return $\perp$, where $r \in \mathcal{R}_\lambda$;
$sk_{f'} \leftarrow \mathsf{S}_2^{\mathsf{KeyIdeal}(\overrightarrow{x},\cdot)}(st', \cdot);$
where $f'(x, r_i)\mathsf{KeyIdeal}(\overrightarrow{x}, f')$; $r_i \leftarrow_\$ \mathcal{R}_\lambda$;
$fS \leftarrow fS \cup \{f'\};\ KGS \leftarrow KGS \cup \{\phi_n\};$
Return $sk_{f'}$.

$\underline{\mathsf{ENC}'(mpk, x, 1, n)}$
If $fq = false$ or $target = false$ or $(x \in \overrightarrow{x})$ or
  $\phi_n \in (LRS \cup KGS)$ or $(mpk \notin \{mpk_i\}_{i\in[l]})$
  then return $\perp$;
$ct \leftarrow \mathsf{rFE.E}(mpk, x; \phi_n(r));$
$ENS \leftarrow ENS \cup \{\phi_n\}$; Return ct.

$\underline{\mathsf{LR}'(\{x_i\}_{i\in[q_{lr}]}, 1, n)}$
If $fq = false$ or $target = false$ or
  $\phi_n \in (ENS \cup DES)$
  then return $\perp$;
For each $i \in [q_{lr}]$
  compute $ct_i^* \leftarrow \mathsf{rFE.E}(mpk^*, 0^{|x_i|}; \phi_n(r));$
  $ctr \leftarrow ctr + 1;$
$LRS \leftarrow LRS \cup \{\phi_n\};\ ChS \leftarrow ChS \cup \{ct_i^*\}_{i\in[q_{lr}]};$
If $ctr = q_{lr}$ then $lrq \leftarrow true$;
Return $\{ct_i^*\}_{i\in[q_{lr}]}$.

$\underline{\mathsf{DEC}'^{\mathsf{DecIdeal}(\cdot,\cdot)}(ct, g', 1, n)}$
If $fq = false$ or $target = false$ or $ct \in ChS$ or
  $\phi_n \in (LRS \cup KGS)$
  then return $\perp$;
$g'(x, r) \leftarrow \mathsf{S}_3^{\mathsf{DecIdeal}(\cdot,\cdot)}(st', \cdot);\ r \leftarrow_\$ \mathcal{R}_\lambda;$
$DES \leftarrow DES \cup \{\phi_n\};\ gS \leftarrow gS \cup \{g'\};$
Return $y' = g'(x, r)$.

$\underline{\mathsf{Finalise}(\alpha)}$
Return $\mathsf{R}(\overrightarrow{x}, fS, gS, \{y'\}, lfS, \alpha)$.

Figure G1 SIM-LR-RRA-CCA??

In addition, note that in the ideal game, KeyGen' denotes the simulator algorithm $\mathsf{S}_2(st', \cdot)$ that has oracle access to the ideal functionality $\mathsf{KeyIdeal}(\overrightarrow{x}, \cdot)$. The functionality $\mathsf{KeyIdeal}$ accepts key queries f' and returns $f'(x_i; r_i)$ for every $x_i \in \overrightarrow{x}$ and $r_i \leftarrow_\$ \mathcal{R}_\lambda$. The set fS denotes the key queries made by S_2 to $\mathsf{KeyIdeal}$. DEC' denotes the simulator algorithm $\mathsf{S}_3(st', \cdot)$ that has oracle access to ideal functionality $\mathsf{DecIdeal}(\cdot, \cdot)$. The functionality $\mathsf{DecIdeal}$ accepts input queries (x, g') and returns $y' = g'(x; r)$ for $r \leftarrow_\$ \mathcal{R}_\lambda$. The set gS denotes the functions that appear in the queries of S_3 and $\{y'\}$ denotes the responses of $\mathsf{DecIdeal}$.

Definition 7 (Sim-LR-RRA-CCA Security for rFE)**.** A functional encryption scheme rFE for randomized function family $\mathcal{F}$ is said to be l_L-leakage-resilient simulation-based RRA and CCA secure (l_L-$\mathsf{Sim\text{-}LR\text{-}RRA\text{-}CCA}$ secure) if for all PPT Φ-restricted adversary $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$, the advantage of the $\mathsf{Sim\text{-}LR\text{-}RRA\text{-}CCA}$ adversary $\mathcal{A}$ against the FE scheme rFE with respect to the l_L-bit leakage, simulator $\mathsf{S}=(\mathsf{S}_1,\mathsf{S}_2,\mathsf{S}_3)$, randomized function family $\mathcal{F}$ and boolean relation R, namely,

$$\mathsf{Adv}^{\mathsf{Sim\text{-}LR\text{-}RRA\text{-}CCA}}_{\mathsf{rFE},l_L,\mathsf{S},\mathcal{F},\mathsf{R},\mathcal{A}}(\lambda)=\Pr[\mathsf{LRRRACCAREAL}^{\mathcal{A}}_{\mathsf{rFE},l_L,\mathcal{F},\mathsf{R}}(\lambda)]-\Pr[\mathsf{LRRRACCAIDEAL}^{\mathsf{S}}_{\mathsf{rFE},l_L,\mathcal{F},\mathsf{R}}(\lambda)],$$

is negligible in the security parameter λ.

Appendix H Security

Theorem 1. Assume that the ϵ_1-universal trapdoor hash proof system THPS exists, let rFE be a public-key functional encryption for randomized function family $\mathcal{F}$. Let LAF be an (l_{LAF},n') lossy algebraic filter, $\mathsf{Ext}:\mathcal{K}\times\{0,1\}^d\rightarrow\{0,1\}^m$ an average-case $((\nu-(q_{lr}-1).m-q_{lr}.l_{\mathsf{LAF}}-l_L),\epsilon_2)$-strong extractor, DIO a secure differing-inputs obfuscator, h an *l-mk-sci-pr* secure CIS hash function, PF a secure puncturable PRF, $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$ a PPT Φ-restricted adversary, l_L a bounded amount of leakage, R a boolean relation and let $\nu-(q_{lr}-1).m-q_{lr}.l_{\mathsf{LAF}}-l_L\geqslant m+\omega(\log\lambda)$ and $\nu=\log(1/\epsilon_1)$, where m is the message length, then there exists a simulator $\mathsf{S}=(\mathsf{S}_1,\mathsf{S}_2,\mathsf{S}_3)$ and adversaries $\mathcal{A}_{cis}$, $\mathcal{A}_{ind}$, $\mathcal{A}_{smp}$, $\mathcal{A}_{dio}$, $\mathcal{A}_{prf}$, $\mathcal{A}'_{prf}$ and $\mathcal{A}_{evs}$ such that

$$\begin{aligned}\mathsf{Adv}^{\mathsf{Sim\text{-}LR\text{-}RRA\text{-}CCA}}_{\mathsf{rFE},l_L,\mathsf{S},\mathcal{F},\mathsf{R},\mathcal{A}}(\lambda)\ &\leqslant\ q_r.q_\phi.\mathsf{Adv}^{l\text{-mk-sci-pr}}_{\mathsf{h},\mathcal{A}_{cis}}(\lambda)+q_r.(q_\phi+q_d).\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PF},\mathcal{A}_{prf}}(\lambda)+q_r.\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LAF},\mathcal{A}_{ind}}(\lambda)\\&+\ q_r.q_{lr}.\mathsf{Adv}^{\mathsf{smp}}_{\mathsf{THPS},\mathcal{A}_{smp}}(\lambda)+q_r.q_{sk}.\mathsf{Adv}^{\mathsf{dio}}_{\mathcal{G}_f,\mathsf{S}.\mathcal{G}_f,\mathcal{A}_{dio}}(\lambda)+q_r.q_{sk}.\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PF},\mathcal{A}'_{prf}}(\lambda)\\&+\ q_r.q_d.\mathsf{Adv}^{\mathsf{evs}}_{\mathsf{LAF},\mathcal{A}_{evs}}(\lambda)+q_r.q_d.2^{l_L+q_{lr}.l_{\mathsf{LAF}}+q_{lr}.m}/(2^\nu-q_d)+2.q_r.q_{lr}.\epsilon_2\\&+\ \frac{l^2.q_r}{|\mathrm{HashKeySpace}|}.\end{aligned}\tag{H1}$$

Adversaries $\mathcal{A}_{cis}$, $\mathcal{A}_{ind}$, $\mathcal{A}_{smp}$, $\mathcal{A}_{dio}$, $\mathcal{A}_{prf}$, $\mathcal{A}'_{prf}$ and $\mathcal{A}_{evs}$ run in approximately the same time as $\mathcal{A}$ which uses at most q_r randomness indices and q_ϕ functions in its oracle queries, and makes at most q_{lr} LR queries, q_{sk} KeyGen queries and q_d DEC queries. *Proof.* In the following, we first give a description about the simulator $\mathsf{S}=(\mathsf{S}_1,\mathsf{S}_2,\mathsf{S}_3)$, then define a sequence of games and prove that the output of every game is computationally indistinguishable from that of its adjacent game. In each game, we assume that the adversary $\mathcal{A}$ makes at most q_d queries to the DEC oracle, q_{sk} queries to the KeyGen oracle, q_e queries to the ENC oracle, q_{lr} queries to the LR oracle and uses at most q_r randomness indices and q_ϕ functions in the KeyGen, LR, ENC and DEC oracle queries altogether.

Description of Simulator.

In the following, we describe a simulator $\mathsf{S}=(\mathsf{S}_1,\mathsf{S}_2,\mathsf{S}_3)$ that makes black-box use of a real world adversary $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$.

Simulator S_1. S_1 performs a simulated setup procedure, namely, for all $i\in[l]$, it first computes $(pp_i,td_i)\leftarrow_\$\mathsf{THPS.Gen}(1^\lambda)$, $(lpk_i,ltk_i)\leftarrow_\$\mathsf{LAF.KG}(1^\lambda)$, $k_i\leftarrow_\$\mathsf{CIH}.\mathcal{K}(1^\lambda)$, $sk_i\leftarrow_\$\mathcal{SK}$, $pk_i=\mu(sk_i)$ and then sets $\{mpk_i=(pp_i,lpk_i,k_i,pk_i)\}_{i\in[l]}$, $\{msk_i=sk_i\}_{i\in[l]}$.

Simulator S_2. S_2 simulates the key generation oracle and the challenge ciphertext generation oracle.

When the adversary $\mathcal{A}_2$ makes a LR' query $(\{|x_i|\}_{i\in[q_{lr}]},1,n)$ priori to the key generation query, the simulator S_2 computes $ct_i^*=(C^*,s^*,U^*,\pi^*,t_c^*)$ as follows. For each $i\in[q_{lr}]$, it first picks a true random $r\leftarrow_\$\mathcal{R}_\lambda$, then samples $(C^*,s^*)\leftarrow Sample_{\mathcal{C}\backslash\mathcal{V}\times\{0,1\}^d}(r)$, and computes $K^*=\mathsf{THPS}(sk^*,C^*)$, $U^*=\mathsf{Ext}(K^*,s^*)\oplus 0^{|x_i|}$, and $\pi^*=\mathsf{LAF}_{lpk^*,t^*}(K^*)$, where $t^*=(t_a^*,t_c^*)$, $t_a^*=(C^*,s^*,U^*)$ and $t_c^*=\mathsf{LAF.Ltag}(ltk^*,t_a^*)$. This can be done since, at this time, the adversary $\mathcal{A}_2$ has finished the Func' and the Target' queries, the simulator has known the target key $mpk^*=(pp^*,lpk^*,k^*,pk^*)$, $msk^*=sk^*$, td^* and ltk^*.

When the adversary $\mathcal{A}_2$ makes KeyGen' query $(f,1,n)$, the simulator S_2 executes the following steps. Note that in this phase, since the adversary $\mathcal{A}_2$ has finished all LR' queries, S_2 has known the resulting challenge ciphertext $\{ct_i^*\}_{i\in[q_{lr}]}$ and thus can proceed the following steps.

1. First query the ideal functionality $\mathsf{KeyIdeal}(\overrightarrow{x},\cdot)$ on input f'. Let $\{y_i^*\}_{i\in[q_{lr}]}$ be the output of $\mathsf{KeyIdeal}(\overrightarrow{x},f')$, namely, for every $i\in[q_{lr}]$, $y_i^*=f'(x_i,r_i)$ with $r_i\leftarrow_\$\mathcal{R}_\lambda$ and $|\overrightarrow{x}|=q_{lr}$.
2. Choose a PRF key $r'\leftarrow_\$\mathcal{R}_\lambda$ and computes the punctured key $r'_{\mathsf{P}}\leftarrow\mathsf{PF.Punc}(r',\{ct_i^*\}_{i\in[q_{lr}]})$.
3. Compute the secret key $sk_{f'}\leftarrow\mathsf{DIO}(\mathsf{S}.\mathcal{G}_{[mpk^*,td^*,r'_{\mathsf{P}},\{ct_i^*\}_{i\in[q_{lr}]},\{y_i^*\}_{i\in[q_{lr}]},f']})$ and return $sk_{f'}$, where the circuit $\mathsf{S}.\mathcal{G}_{[mpk^*,td^*,r'_{\mathsf{P}},\{ct_i^*\}_{i\in[q_{lr}]},\{y_i^*\}_{i\in[q_{lr}]},f']}$ is constructed in Figure H1.

Simulator S_3. The simulator S_3 simulates the decryption oracle. In fact, in this phase, S_3 has got the state information from simulator S_2, therefore S_3 can proceed the following steps smoothly. When $\mathcal{A}_2$ makes a decryption query $(ct,g',1,n)$ with $ct=(C,s,U,\pi,t_c)$, the simulator S_3 performs the following steps.

1. If $C\in\mathcal{C}\backslash\mathcal{V}$ or $ct\in\{ct_i^*\}_{i\in[q_{lr}]}$ ($C\in\mathcal{C}\backslash\mathcal{V}$ can be decided with the corresponding universal trapdoor), then output $\perp$ and stop, otherwise continue the next step.
2. Compute $K=\mathsf{THPS.Priv}(sk^*,C)$ and $\pi'=\mathsf{LAF}_{lpk^*,t}(K)$, if $\pi'\neq\pi$, output $\perp$ and stop, otherwise continue the next step.
3. Compute $x=\mathsf{Ext}(K,s)\oplus U$.
4. Return $g'(x;r)=\mathsf{DecIdeal}(g',x)$ with $r\leftarrow_\$\mathcal{R}_\lambda$.

The sequence of games.

G_0 : This is the real experiment. Here, each decryption query $(ct, g, 1, n)$ is answered using a decryption key $sk_g \leftarrow \mathsf{DIO}(\mathcal{G}_{[mpk^*,msk^*,r',g]})$ where $\mathcal{G}_{[mpk^*,msk^*,r',g]}$ is defined in the same manner as $\mathcal{G}_{[mpk^*,msk^*,r',f]}$, except that it has function g hardcoded in it. Let $\{ct_i^* = (C^*, s^*, U^*, \pi^*, t_c^*)\}_{i\in[q_{lr}]}$ be challenge ciphertext set in which ct_i^* encrypts $x_i \in \overrightarrow{x}$ where $|\overrightarrow{x}| = l_{lr}$. For convenience, we write the circuit $\mathcal{G}_{[mpk^*,msk^*,r',g]}$ as $\mathcal{G}_g$ which is shown in Figure **??**.

G_1 : This game is the same as G_0 except for the queries on the target key. Rather than using the hash values $\mathsf{h}_{k^*}(\phi_n(r))$, the game picks a uniformly random value from $\mathcal{R}_\lambda$ to replace $\mathsf{h}_{k^*}(\phi_n(r))$. Concretely, in the KeyGen queries, the secret key sk_f is computed as $sk_f \leftarrow \mathsf{DIO}(\mathcal{G}_{[mpk^*,msk^*,r',f]})$ with $r' = \mathsf{PF}_{r_\mathsf{h}}(mpk||f)$, where r_h is chosen uniformly and randomly from $\mathcal{R}_\lambda$, instead of $r_\mathsf{h} = \mathsf{h}_{k^*}(\phi_n(r))$. The values $\mathsf{h}_{k^*}(\phi_n(r))$ in the LR and ENC queries are also replaced with uniformly random values. While in the DEC query, the same replacement is done as in KeyGen queries. In fact, the gaps between games G_0 and G_1 may be reduced to the security of the CIS hash function h. Note that in order to prevent trivial attacks, we require that the functions ϕ_n used in the KeyGen query must be different from each other and cannot appear in other oracle queries.

G_2 : This game is the same as G_1 except for the queries on the target key. Rather than using $\mathsf{PF}_{r_\mathsf{h}}(\cdot)$, the game picks a uniformly random value from $\mathcal{R}_\lambda$ to replace $\mathsf{PF}_{r_\mathsf{h}}(\cdot)$. Concretely, in the KeyGen queries, the secret key sk_f is computed as $sk_f \leftarrow \mathsf{DIO}(\mathcal{G}_{[mpk^*,msk^*,r',f]})$, where r' is chosen uniformly and randomly from $\mathcal{R}_\lambda$, instead of $r' = \mathsf{PF}_{r_\mathsf{h}}(mpk||f)$. The value $\mathsf{PF}_{r_\mathsf{h}}(mpk||x_i)$ in the LR queries and the value $\mathsf{PF}_{r_\mathsf{h}}(mpk||x)$ in the ENC queries are also replaced with uniformly random values. While in the DEC query, the same replacement is done as in KeyGen queries. In fact, the gaps between games G_1 and G_2 may be reduced to the security of the puncturable PRF PF.

G_3 : This game is the same as G_2 with the exception that the generation of the core tag t_c^* in every challenge ciphertext in the set $\{ct_i^*\}_{i\in[q_{lr}]}$ is computed as $t_c^* = \mathsf{LAF.Ltag}(ltd^*, t_a^*)$ instead of sampling t_c^* randomly and uniformly from $\mathcal{T}_c$, where $t_a^* = (C^*, s^*, U^*)$.

G_4 : This game is the same as G_3 except for the computation of K^* in every challenge ciphertext in the set $\{ct_i^*\}_{i\in[q_{lr}]}$. This game computes $K^* = \mathsf{THPS.Priv}(sk^*, C^*)$ rather than $K^* = \mathsf{THPS.Pub}(pk^*, C^*, w^*)$.

G_5 : This game is the same as G_4 except for the generation of C^* in every challenge ciphertext in the set $\{ct_i^*\}_{i\in[q_{lr}]}$. We samples $C^* \leftarrow_\$ \mathcal{C}\backslash\mathcal{V}$ instead of $C^* \leftarrow_\$ \mathcal{V}$.

G_6 : This game is the same as G_5 except that for every key query for function f, the secret key is answered with $sk_f \leftarrow \mathsf{DIO}(\mathsf{S}.\mathcal{G}_{[mpk^*,td^*,r'_\mathsf{p},\{ct_i^*\}_{i\in[q_{lr}]},\{y_i^*\}_{i\in[q_{lr}]},f]})$, where $y_i^* = f(x_i; r_i)$ with $r_i = \mathsf{PF.Eval}(r', ct_i^*)$ and $r'_\mathsf{p} = \mathsf{PF.Punc}(r', \{ct_i^*\}_{i\in[q_{lr}]})$ and $\mathsf{S}.\mathcal{G}_{[mpk^*,td^*,r'_\mathsf{p},\{ct_i^*\}_{i\in[q_{lr}]},\{y_i^*\}_{i\in[q_{lr}]},f]}$ is constructed in Figure H1. Similarly, we write the circuit $\mathsf{S}.\mathcal{G}_{[mpk^*,td^*,r'_\mathsf{p},\{ct_i^*\}_{i\in[q_{lr}]},\{y_i^*\}_{i\in[q_{lr}]},f]}$ as $\mathsf{S}.\mathcal{G}_f$.

G_7 : This game is the same as G_6 except that for every key query for function f, the secret key is answered with $sk_f \leftarrow \mathsf{DIO}(\mathsf{S}.\mathcal{G}_{[mpk^*,td^*,r'_\mathsf{p},\{ct_i^*\}_{i\in[q_{lr}]},\{y_i^*\}_{i\in[q_{lr}]},f]})$ in the same manner as the simulator S_2, where $y_i^* = f(x_i; r_i)$ with $r_i \leftarrow_\$ \mathcal{R}_\lambda$.

G_8 : This game is the same as G_7 except that when the adversary delivers a decryption query $(ct, g, 1, n)$ such that $ct = (C, s, U, \pi, t_c)$ with $C \in \mathcal{C}\backslash\mathcal{V}$, the decryption oracle outputs $\perp$.

G_9 : This game is the same as G_8 except that U^* in every challenge ciphertext in the set $\{ct_i^*\}_{i\in[q_{lr}]}$ is computed as $U^* = \mathsf{Ext}(K^*, s^*) \oplus 0^{|x_i|}$ instead of $U^* = \mathsf{Ext}(K^*, s^*) \oplus x_i$.

G_{10} : This game is the same as G_9 except that we now answer the decryption queries of $\mathcal{A}_2$ in the same manner as simulator S_3. Note that this is the ideal experiment.

Obviously, in game G_0, we are in an identical setting to the real game in the Sim-LR-RRA-CCA security definition, while in game G_{10}, we are in an identical setting to the ideal game. Let coll denotes the event that collisions happen in the hash function keys, then the advantage of a Sim-LR-RRA-CCA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ can be defined as

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{Sim\text{-}LR\text{-}RRA\text{-}CCA}}_{\mathsf{rFE},l_L,\mathsf{S},\mathcal{F},\mathsf{R},\mathcal{A}}(\lambda) &= \mathsf{Pr}[\mathsf{LRRRACCAREAL}^{\mathcal{A}}_{\mathsf{rFE},l_L,\mathcal{F},\mathsf{R}}(\lambda)] - \mathsf{Pr}[\mathsf{LRRRACCAIDEAL}^{\mathsf{S}}_{\mathsf{rFE},l_L,\mathcal{F},\mathsf{R}}(\lambda)] \\
&= \Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_{10}] \\
&\leqslant \Pr[\mathsf{G}_0|\overline{\mathsf{coll}}] - \Pr[\mathsf{G}_{10}|\overline{\mathsf{coll}}] + \Pr[\mathsf{coll}] \\
&= \sum_{i=0}^{9}(\Pr[\mathsf{G}_i|\overline{\mathsf{coll}}] - \Pr[\mathsf{G}_{i+1}|\overline{\mathsf{coll}}]) + \Pr[\mathsf{coll}] \qquad \text{(H2)}
\end{aligned}$$

For simplicity, we invoke Lemma 1 from [5], so that we now only have to prove the theorem for an adversary using just one random value. In the following, we first give the descriptions of a series of Claims, then prove them in Appendix 10.

Lemma 1. Let $\Pr[\mathsf{G}_0|\overline{\mathsf{coll}}]$ and $\Pr[\mathsf{G}_1|\overline{\mathsf{coll}}]$ respectively denote the probability that adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ outputs 1 in game G_0 and game G_1 when no collisions happen in the hash function keys. Then we can construct an adversary $\mathcal{A}_{cis}$ which tries to break the l-mk-sci-pr security of the CIS hash function h such that

$$\Pr[\mathsf{G}_0|\overline{\mathsf{coll}}] - \Pr[\mathsf{G}_1|\overline{\mathsf{coll}}] \leqslant q_\phi.\mathsf{Adv}^{l\text{-mk-sci-pr}}_{\mathsf{h},\mathcal{A}_{cis}}(\lambda). \qquad \text{(H3)}$$

Lemma 2. Let $\Pr[\mathsf{G}_1|\overline{\mathsf{coll}}]$ and $\Pr[\mathsf{G}_2|\overline{\mathsf{coll}}]$ respectively denote the probability that adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ outputs 1 in game G_1 and game G_2 when no collisions happen in the hash function keys. Then we can construct an adversary $\mathcal{A}_{prf}$ which tries to break the security of the puncturable PRF PF such that

$$\Pr[\mathsf{G}_1|\overline{\mathsf{coll}}] - \Pr[\mathsf{G}_2|\overline{\mathsf{coll}}] \leqslant q_\phi.\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PF},\mathcal{A}_{prf}}(\lambda). \qquad \text{(H4)}$$

Constants : mpk^*, td^*, r'_{p}, $\{ct_i^*\}_{i\in[q_{lr}]}$, $\{y_i^*\}_{i\in[q_{lr}]}$, f
Input : ct

1. If $ct = ct_i^* \in \{ct_i^*\}_{i\in[q_{lr}]}$ output y_i^* and stop;
2. Parse ct into C, s, U, π, t_c and parse mpk^* into pp^*, lpk^*, k^*, pk^*.
3. If $C \in \mathcal{C}\backslash\mathcal{V}$ output $\perp$ and stop.
4. Compute $w = \mathsf{THPS.Invert}(td^*, C)$.
5. Compute $K = \mathsf{THPS.Pub}(pk^*, C, w)$.
6. Compute $\pi' = \mathsf{LAF}_{lpk^*,t}(K)$, where $t = (t_a, t_c)$, $t_a = (C, s, U)$.
7. If $\pi' \neq \pi$, output $\perp$ and stop, else proceed the following steps.
8. Compute $x = \mathsf{Ext}(K, s) \oplus U$.
9. Compute $r'' = \mathsf{PF.Eval}(r'_{\mathsf{p}}, ct)$.
10. Compute $y = f(x; r'')$ and output y.

Figure H1 Functionality $\mathsf{S}.\mathcal{G}_{[mpk^*,td^*,r'_{\mathsf{p}},\{ct_i^*\}_{i\in[q_{lr}]},\{y_i^*\}_{i\in[q_{lr}]},f]}$

Lemma 3. Let $\Pr[\mathsf{G}_2|\overline{\mathsf{coll}}]$ and $\Pr[\mathsf{G}_3|\overline{\mathsf{coll}}]$ respectively denote the probability that adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ outputs 1 in game G_2 and game G_3 when no collisions happen in the hash function keys. Then we can construct an adversary $\mathcal{A}_{ind}$ which tries to break the indistinguishability property of the lossy algebraic filter LAF such that

$$\Pr[\mathsf{G}_2|\overline{\mathsf{coll}}] - \Pr[\mathsf{G}_3|\overline{\mathsf{coll}}] \leqslant \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LAF},\mathcal{A}_{ind}}(\lambda). \tag{H5}$$

Lemma 4. Let $\Pr[\mathsf{G}_3|\overline{\mathsf{coll}}]$ and $\Pr[\mathsf{G}_4|\overline{\mathsf{coll}}]$ respectively denote the probability that adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ outputs 1 in game G_3 and game G_4 when there are no collisions in the hash function keys. Then we have

$$\Pr[\mathsf{G}_3|\overline{\mathsf{coll}}] = \Pr[\mathsf{G}_4|\overline{\mathsf{coll}}]. \tag{H6}$$

Lemma 5. Let $\Pr[\mathsf{G}_4|\overline{\mathsf{coll}}]$ and $\Pr[\mathsf{G}_5|\overline{\mathsf{coll}}]$ respectively denote the probability that adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ outputs 1 in game G_4 and game G_5 when there are no collisions in the hash function keys. Then we can construct an adversary $\mathcal{A}_{smp}$ which can break the subset membership problem of the trapdoor hash proof system THPS such that

$$\mathsf{Pr}[\mathsf{G}_4|\overline{\mathsf{coll}}] - \mathsf{Pr}[\mathsf{G}_5|\overline{\mathsf{coll}}] \leqslant q_{lr}.\mathsf{Adv}^{\mathsf{smp}}_{\mathsf{THPS},\mathcal{A}_{smp}}(\lambda). \tag{H7}$$

Lemma 6. Let $\Pr[\mathsf{G}_5|\overline{\mathsf{coll}}]$ and $\Pr[\mathsf{G}_6|\overline{\mathsf{coll}}]$ respectively denote the probability that adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ outputs 1 in game G_5 and game G_6 when there are no collisions in the hash function keys. Then we can construct an adversary $\mathcal{A}_{dio}$ which tries to break the security of the differing-inputs obfuscation such that

$$\Pr[\mathsf{G}_5|\overline{\mathsf{coll}}] - \Pr[\mathsf{G}_6|\overline{\mathsf{coll}}] \leqslant q_{sk}.\mathsf{Adv}^{\mathsf{dio}}_{\mathcal{G}_f,\mathsf{S}.\mathcal{G}_f,\mathcal{A}_{dio}}(\lambda). \tag{H8}$$

Lemma 7. Let $\Pr[\mathsf{G}_6|\overline{\mathsf{coll}}]$ and $\Pr[\mathsf{G}_7|\overline{\mathsf{coll}}]$ respectively denote the probability that adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ outputs 1 in game G_6 and game G_7 when there are no collisions in the hash function keys. Then we can construct an adversary $\mathcal{A}'_{prf}$ which tries to break the security of the puncturable PRF PF such that

$$\Pr[\mathsf{G}_6|\overline{\mathsf{coll}}] - \Pr[\mathsf{G}_7|\overline{\mathsf{coll}}] \leqslant q_{sk}.\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PF},\mathcal{A}'_{prf}}(\lambda). \tag{H9}$$

Lemma 8. Let $\Pr[\mathsf{G}_7|\overline{\mathsf{coll}}]$ and $\Pr[\mathsf{G}_8|\overline{\mathsf{coll}}]$ respectively denote the probability that adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ outputs 1 in game G_7 and game G_8 when there are no collisions in the hash function keys. Then we can construct an adversary $\mathcal{A}_{evs}$ which can break the evasiveness security of the lossy algebraic filter LAF such that

$$\mathsf{Pr}[\mathsf{G}_7|\overline{\mathsf{coll}}] - \mathsf{Pr}[\mathsf{G}_8|\overline{\mathsf{coll}}] \leqslant q_d.\mathsf{Adv}^{\mathsf{evs}}_{\mathsf{LAF},\mathcal{A}_{evs}}(\lambda) + q_d.2^{l_L + q_{lr}.l_{\mathsf{LAF}} + q_{lr}.m}/(2^{\nu} - q_d). \tag{H10}$$

Lemma 9. Let $\Pr[\mathsf{G}_8|\overline{\mathsf{coll}}]$ and $\Pr[\mathsf{G}_9|\overline{\mathsf{coll}}]$ respectively denote the probability that adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ outputs 1 in game G_8 and game G_9 when no collisions happen in the hash function keys. Then we have

$$\mathsf{Pr}[\mathsf{G}_8|\overline{\mathsf{coll}}] - \mathsf{Pr}[\mathsf{G}_9|\overline{\mathsf{coll}}] \leqslant 2.q_{lr}.\epsilon_2. \tag{H11}$$

Lemma 10. Let $\Pr[\mathsf{G}_9|\overline{\mathsf{coll}}]$ and $\Pr[\mathsf{G}_{10}|\overline{\mathsf{coll}}]$ respectively denote the probability that adversary $\mathcal{A}$ outputs 1 in game G_9 and game G_{10} when there are no collisions in the hash function keys. Then we can construct an adversary $\mathcal{A}_{prf}$ which can break the security of the puncturable PRF PF such that

$$\Pr[\mathsf{G}_9|\overline{\mathsf{coll}}] - \Pr[\mathsf{G}_{10}|\overline{\mathsf{coll}}] \leqslant q_d.\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PF},\mathcal{A}_{prf}}(\lambda). \tag{H12}$$

We complete the description of these Claims. We give their proofs in Appendix 10. If let |HashKeySpace| denote the size of the hash key space, then obviously we have $\mathsf{Pr}[\mathsf{coll}] \leqslant \frac{l^2}{|\mathrm{HashKeySpace}|}$. In addition, since the adversary is allowed to query at most q_r randomness indices in oracle queries, by Eq. H2 and Eqs. H3, H4, H5, H6, H7, H8, H9, H10, H11 and H12, we get Eq. H1, namely,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{Sim\text{-}LR\text{-}RRA\text{-}CCA}}_{\mathsf{rFE},l_L,\mathsf{S},\mathcal{F},\mathsf{R},\mathcal{A}}(\lambda) &= \mathsf{Pr}[\mathsf{LRRRACCAREAL}^{\mathcal{A}}_{\mathsf{rFE},l_L,\mathcal{F},\mathsf{R}}(\lambda)] - \mathsf{Pr}[\mathsf{LRRRACCAIDEAL}^{\mathsf{S}}_{\mathsf{rFE},l_L,\mathcal{F},\mathsf{R}}(\lambda)] \\ &\leqslant \sum_{i=0}^{9}(\Pr[\mathsf{G}_i|\overline{\mathsf{coll}}] - \Pr[\mathsf{G}_{i+1}|\overline{\mathsf{coll}}]) + \Pr[\mathsf{coll}] \\ &\leqslant q_r.q_\phi.\mathsf{Adv}^{l\text{-mk-sci-pr}}_{\mathsf{h},\mathcal{A}_{cis}}(\lambda) + q_r.(q_\phi+q_d).\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PF},\mathcal{A}_{prf}}(\lambda) + q_r.\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LAF},\mathcal{A}_{ind}}(\lambda) \\ &+ q_r.q_{lr}.\mathsf{Adv}^{\mathsf{smp}}_{\mathsf{THPS},\mathcal{A}_{smp}}(\lambda) + q_r.q_{sk}.\mathsf{Adv}^{\mathsf{dio}}_{\mathcal{G}_f,\mathsf{S}.\mathcal{G}_f,\mathcal{A}_{dio}}(\lambda) + q_r.q_{sk}.\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PF},\mathcal{A}'_{prf}}(\lambda) \\ &+ q_r.q_d.\mathsf{Adv}^{\mathsf{evs}}_{\mathsf{LAF},\mathcal{A}_{evs}}(\lambda) + q_r.q_d.2^{l_L+q_{lr}.l_{\mathsf{LAF}}+q_{lr}.m}/(2^\nu - q_d) + 2.q_r.q_{lr}.\epsilon_2 \\ &+ \frac{l^2.q_r}{|\mathrm{HashKeySpace}|}.\end{aligned}$$

This proves Theorem 1.

Completing Proof of Theorem 1

Here we give the complete proofs for Claim 1 to Claim 10 described above.

Proof of Claim 1 . If no collisions happen in the hash function keys, then the difference between G_0 and G_1 can be reduced to the security of the CIS hash function h. That is, if there exists an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ can distinguish G_0 and G_1, we can construct an adversary $\mathcal{A}_{cis}$ which uses $\mathcal{A}$ to break the security of the CIS hash function h. By a hybrid argument, the reductions can be perfectly completed in the same way as Lemma 2 in [5] by the adversary $\mathcal{A}_{cis}$ and thus, we have equation H3 hold.

Proof of Claim 2 . If no collisions happen in the hash function keys, obviously the difference between G_1 and G_2 can be reduced to the security of the puncturable PRF PF. That is, if there exists an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ can distinguish G_1 and G_2, we can construct an adversary $\mathcal{A}_{prf}$ which uses $\mathcal{A}$ to break the security of the puncturable PRF PF. In the same way as Lemma 3 in [5], by a hybrid argument, the reductions can be perfectly simulated by the adversary $\mathcal{A}_{prf}$ and thus, we have equation H4 hold.

Proof of Claim 3 . We show that when there are no collisions in the hash keys, if there exists a PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that can distinguish game G_2 and game G_3, then we can build a PPT adversary $\mathcal{A}_{ind}$ that can break the indistinguishability (IND) security of the lossy algebraic filter LAF. Let $\mathcal{B}$ denote the IND challenger of the LAF, then the adversary $\mathcal{A}_{ind}$ is constructed as follows.

1. The adversary $\mathcal{A}_{ind}$ first honestly generates $(\{mpk_{i'}\}_{i'\in[l]}, \{msk_{i'}\}_{i'\in[l]}, st')$ except that it gets $\{lpk_{i'}\}_{i'\in[l]}$ from its IND challenger $\mathcal{B}$. Then it flips a coin $b \leftarrow_\$ \{0,1\}$. Note that although the adversary $\mathcal{A}_{ind}$ may obtain the trapdoor $td_{i'}$, since the generation of $td_{i'}$ is independent of LAF, it would not help the adversary $\mathcal{A}_{ind}$ to win in this reduction.
2. For every LR query from $\mathcal{A}_2$, $\mathcal{A}_{ind}$ first constructs $t_a^* = (C^*, s^*, U^*)$ by itself and then forwards t_a^* to its challenger $\mathcal{B}$ and receives t_c^* which is either chosen uniformly and randomly from $\mathcal{T}_c$ or computed as $\mathsf{LAF.Ltag}(ltk^*, t_a^*)$, where $(C^*, s^*) \leftarrow Sample_{\mathcal{V}\times\{0,1\}^d}(r)$ with $r \leftarrow_\$ \mathcal{R}_\lambda$. Next $\mathcal{A}_{ind}$ computes $K^* = \mathsf{THPS.Pub}(pk^*, C^*, w^*)$ and $\pi^* = \mathsf{LAF}_{lpk^*,t^*}(K^*)$, where $t^* = (t_a^*, t_c^*)$. Finally $\mathcal{A}_{ind}$ sends the challenge ciphertext $ct^* = (C^*, s^*, U^*, \pi^*, t_c^*)$ to the adversary $\mathcal{A}_2$. Note that all the simulations in this phase can be done, since, priori to the LR query, both the Func query and Target query have been completed in advance.
3. For every ENC query from $\mathcal{A}_2$ such that $(mpk, x, 1, \phi)$, where $mpk = (pp_{i'}, lpk_{i'}, k_{i'}, pk_{i'}) \in \{mpk_{i'}\}_{i'\in[l]}$, since t_c is chosen uniformly and randomly, $\mathcal{A}_{ind}$ can construct the ciphertext $ct = (C, s, U, \pi, t_c)$ by itself as in game G_2 and game G_3. Namely, if $mpk = mpk^*$, then $(C, s, t_c) \leftarrow Sample_{\mathcal{V}\times\{0,1\}^d\times\mathcal{T}_c}(r)$ with $r \leftarrow_\$ \mathcal{R}_\lambda$; otherwise, if $mpk \neq mpk^*$, then $(C, s, t_c) \leftarrow Sample_{\mathcal{V}\times\{0,1\}^d\times\mathcal{T}_c}(\mathsf{PF}_{\mathsf{h}_{k^*}(\phi_n(r))}(mpk||f))$ with $r \leftarrow_\$ \mathcal{R}_\lambda$.
4. $\mathcal{A}_{ind}$ continues to simulate the rest of the experiment in the same manner as in G_2 and G_3.
5. Finally, $\mathcal{A}_{ind}$ sends the output of the experiment to $\mathcal{A}_2$ and returns its results to $\mathcal{B}$.

Now if $\mathcal{B}$ returns $t_c^* = \mathsf{LAF.Ltag}(ltk^*, t_a^*)$, then $\mathcal{A}_{ind}$ perfectly simulates game G_3 for $\mathcal{A}_2$, else it simulates game G_2 for $\mathcal{A}_2$. Thus, we have

$$\Pr[\mathsf{G}_2|\overline{\mathsf{coll}}] - \Pr[\mathsf{G}_3|\overline{\mathsf{coll}}] \leqslant \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LAF},\mathcal{A}_{ind}}(\lambda).$$

Proof of Claim 4 . In game G_3, K^* is computed as $K^* = \mathsf{THPS.Pub}(pk^*, C^*, w^*)$, while in game G_4, the computation of K^* is replaced with $K^* = \mathsf{THPS.Priv}(sk^*, C^*)$. When no collisions happen in the hash keys, by the projectiveness of the trapdoor hash proof system THPS, this conversion is equivalent and thus we have

$$\mathsf{Pr}[\mathsf{G}_3|\overline{\mathsf{coll}}] = \mathsf{Pr}[\mathsf{G}_4|\overline{\mathsf{coll}}].$$

Proof of Claim 5 . Assume that the adversary makes a total of q_{lr} LR queries. We consider q_{lr} intermediate hybrids $\mathsf{G}_{4,i}$ for $0 \leqslant i \leqslant q_{lr}$. In $\mathsf{G}_{4,i}$, we respond to the first $q_{lr} - i$ LR queries as in game G_4 and the remaining i LR queries as in G_5. We show that if there exists a PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that can distinguish game $\mathsf{G}_{4,i}$ and game $\mathsf{G}_{4,i+1}$, then we can build a PPT adversary $\mathcal{A}_{smp}$ that can break the subset membership problem of the trapdoor hash proof system THPS. The construction of $\mathcal{A}_{smp}$ is as follows.

1. The adversary $\mathcal{A}_{smp}$ first honestly generates $(\{mpk_{i'}\}_{i'\in[l]}, \{msk_{i'}\}_{i'\in[l]}, st')$ where for every $i' \in [l]$, $pp_{i'}$ in $mpk_{i'}$ comes from the THPS challenger $\mathcal{B}$ and $msk_{i'} = sk_{i'}$ is chosen uniformly and randomly from the set $\mathcal{SK}_{i'}$. Note that the adversary $\mathcal{A}_{smp}$ does not get the trapdoor $td_{i'}$ which is generated and maintained secretly by the challenger $\mathcal{B}$.
2. For the first $q_{lr} - i - 1$ LR queries from $\mathcal{A}_2$, $\mathcal{A}_{smp}$ responds in the same manner as in game G_4. For the last i LR queries, $\mathcal{A}_{smp}$ responds in the same manner as in game G_5.
3. For the $(q_{lr} - i)'$th LR query, $\mathcal{A}_{smp}$ first gets C^* from his challenger $\mathcal{B}$, where C^* denotes either a sampled value from $\mathcal{V}$ or a sampled value from $\mathcal{C}\backslash\mathcal{V}$. It then constructs the rest parts of the challenge ciphertext ct^* as in game G_4. Finally $\mathcal{A}_{smp}$ sends the challenge ciphertext $ct^* = (C^*, s^*, U^*, \pi^*, t_c^*)$ to the adversary $\mathcal{A}_2$.
4. $\mathcal{A}_{smp}$ continues to simulate the rest of the experiment in the same manner as in G_4 and G_5.
5. Finally $\mathcal{A}_{smp}$ sends the output of the experiment to $\mathcal{A}_2$ and returns its results to $\mathcal{B}$.

Now if $\mathcal{B}$ returns $C^* \leftarrow_\$ Sample_{\mathcal{V}}(r)$, then $\mathcal{A}_{smp}$ perfectly simulates game $\mathsf{G}_{4,i}$ for $\mathcal{A}_2$, else it simulates game $\mathsf{G}_{4,i+1}$ for $\mathcal{A}_2$. Where r is chosen uniformly and randomly from $\mathcal{R}_\lambda$ by the challenger $\mathcal{B}$. Thus, by a hybrid argument, we have

$$\Pr[\mathsf{G}_4|\overline{\mathsf{coll}}] - \Pr[\mathsf{G}_5|\overline{\mathsf{coll}}] \leqslant q_{lr}.\mathsf{Adv}^{\mathsf{smp}}_{\mathsf{THPS},\mathcal{A}_{smp}}(\lambda).$$

Note that we assume that all the Claims in the following implicitly contain the condition that no collisions happen in the hash function keys. We would not stress it any more in the proofs below.

Proof of Claim 6 . Note that the only difference between G_5 and G_6 is that in the former, we output $\mathsf{DIO}(\mathcal{G}_f)$ as the key for function f, while in the latter, we output $\mathsf{DIO}(\mathsf{S}.\mathcal{G}_f)$ as the key for function f. In order to prove that the two hybrids are computationally indistinguishable, we first show that the circuit family $(\mathcal{G}_f, \mathsf{S}.\mathcal{G}_f, z, \mathsf{Sam})$ is a differing-inputs distribution, where $(\mathcal{G}_f, \mathsf{S}.\mathcal{G}_f, z)$ is sampled by algorithm $\mathsf{Sam}(1^\lambda)$ and z is an auxiliary input. Then by the security of differing-inputs obfuscation, we would have that $\mathsf{DIO}(\mathcal{G}_f)$ and $\mathsf{DIO}(\mathsf{S}.\mathcal{G}_f)$ are computationally indistinguishable, which in turn would imply that G_5 and G_6 are computationally indistinguishable.

In fact, as long as we can prove that there exists no PPT adversary who can efficiently find a ciphertext ct which makes $\mathcal{G}_f(ct) \neq \mathsf{S}.\mathcal{G}_f(ct)$ hold with non-negligible probability, then the circuit family $(\mathcal{G}_f, \mathsf{S}.\mathcal{G}_f, z, \mathsf{Sam})$ is a differing-inputs distribution. In the following, we give the formal proofs in two cases, i.e., $ct \in \{ct_j^*\}_{j\in[q_{lr}]}$ and $ct \notin \{ct_j^*\}_{j\in[q_{lr}]}$.

• In the first case, for each $ct = ct_j^* = (C_j^*, s_j^*, U_j^*, \pi_j^*, t_{c,j}^*) \in \{ct_j^*\}_{j\in[q_{lr}]}$, the circuit $\mathcal{G}_f$ computes the value ${K_j^*}' = \mathsf{THPS.Priv}(sk^*, C_j^*)$ in the same manner as its previously generated appearance, so we have $\pi_j^* = {\pi_j^*}' = \mathsf{LAF.Eval}_{lpk^*, t_j^*}({K_j^*}')$, where $t_j^* = ((C_j^*, s_j^*, U_j^*), t_{c,j}^*)$. Thus the circuit $\mathcal{G}_f$ outputs $y_j = f(x_j; r_j'')$ where $r_j'' = \mathsf{PF.Eval}(r', ct_j^*)$. Since the circuit $\mathsf{S}.\mathcal{G}_f$ has the hardcoded challenge ciphertexts $\{ct_j^*\}_{j\in[q_{lr}]}$ and values $\{y_j^*\}_{j\in[q_{lr}]}$, when its input $ct = ct_j^* = (C_j^*, s_j^*, U_j^*, \pi_j^*, t_{c,j}^*) \in \{ct_j^*\}_{j\in[q_{lr}]}$, the hardcoded value y_j^* is directly output by this circuit, where $y_j^* = f(x_j; r_j'')$ and $r_j'' = \mathsf{PF.Eval}(r', ct_j^*)$, hence the two circuits have the identical-output behavior at all points $ct \in \{ct_j^*\}_{j\in[q_{lr}]}$.

• In the second case, if $ct = (C, s, U, \pi, t_c)$ is a ciphertext such that $C \in \mathcal{V}$, then there exists a witness w such that $\mathsf{THPS.Priv}(sk^*, C) = \mathsf{THPS.Priv}(pk^*, C, w)$ holds, thus both circuits either output the same $\perp$ or the same non-$\perp$. However, when $C \in \mathcal{C}\backslash\mathcal{V}$, there may exist some points $ct = (C, s, U, \pi, t_c)$ which make the circuit $\mathcal{G}_f$ output non-$\perp$, while the circuit $\mathsf{S}.\mathcal{G}_f$ output $\perp$. In this case, we show that given $(\mathcal{G}_f, \mathsf{S}.\mathcal{G}_f, z, \mathsf{Sam})$, there exists no PPT adversary who can efficiently find a $ct = (C, s, U, \pi, t_c)$ with $C \in \mathcal{C}\backslash\mathcal{V}$ that makes $\mathcal{G}_f(ct) \neq \mathsf{S}.\mathcal{G}_f(ct)$ with non-negligible probability. We now formalize this case.

Assume that there exists an adversary $\mathcal{A}'$ against the differing-inputs of the above circuit family $(\mathcal{G}_f, \mathsf{S}.\mathcal{G}_f, z, \mathsf{Sam})$ which receives as input $(\mathcal{G}_f, \mathsf{S}.\mathcal{G}_f, z)$ and outputs a ciphertext $ct = (C, s, U, \pi, t_c)$ such that $\mathcal{G}_f(ct) \neq \mathsf{S}.\mathcal{G}_f(ct)$. In other words, the adversary can efficiently find a ciphertext $ct = (C, s, U, \pi, t_c)$ such that $C \in \mathcal{C}\backslash\mathcal{V}$ and $\pi = \mathsf{LAF}_{lpk^*,t}(\mathsf{THPS.Priv}(sk^*, C))$ which makes the circuit $\mathcal{G}_f$ output non-$\perp$, while the circuit $\mathsf{S}.\mathcal{G}_f$ outputs $\perp$. In the following, we analyze the probability that the adversary $\mathcal{A}'$ finds such a ct.

Let a ciphertext $ct = (C, s, U, \pi, t_c)$ such that $C \in \mathcal{C}\backslash\mathcal{V}$ and $\pi = \mathsf{LAF}_{lpk^*,t}(\mathsf{THPS.Priv}(sk^*, C))$ denote a bad ciphertext, a tag t with $t = t^*$ denote a repeated tag, bad_{ct} denote the event that $\mathcal{A}'$ finds a bad ciphertext and T denote the event that there exists a $ct = (C, s, U, \pi, t_c)$ with $t = ((C, s, U), t_c)$ being a non-injective, non-repeated tag, then we have

$$\Pr[\mathsf{bad}_{ct}] = \Pr[\mathsf{bad}_{ct} \wedge \overline{T}] + \Pr[\mathsf{bad}_{ct} \wedge T] \leqslant \Pr[\mathsf{bad}_{ct}|\overline{T}] + \Pr[T].$$

Clearly, if the event T happens, it means that there exists a PPT adversary $\mathcal{A}_{evs}$ who can efficiently outputs a non-injective, non-repeated tag $t = ((C, s, U), t_c)$ such that t_c is never queried to its oracle. Assume that the adversary $\mathcal{A}'$ does a total of q_{bad} searches, then by the evasiveness security of the lossy algebraic filter LAF, we have

$$\Pr[T] \leqslant q_{bad}.\mathsf{Adv}^{\mathsf{evs}}_{\mathsf{LAF},\mathcal{A}_{evs}}(\lambda). \tag{H13}$$

Without loss of generality, assume that $ct = (C, s, U, \pi, t_c)$ is the first bad ciphertext that makes the event bad_{ct} happen conditioned on $\overline{T}$. I.e., both $C \in \mathcal{C}\backslash\mathcal{V}$ and $\pi = \mathsf{LAF}_{lpk^*,t}(\mathsf{THPS.Priv}(sk^*, C))$ hold and $t = ((C, s, U), t_c)$ is an injective tag. Let $z =(\{mpk_i\}_{i\in[l]}$, C, $\{ct_i\}_{i\in[q_e]}$, l_L-$leak$, $\{ct_i^*\}_{i\in[q_{lr}]}$, $\mathsf{Leak}(\{sk_f\}_{f\in\{f\}}))$ denote the auxiliary input given to $\mathcal{A}'$, then we have

$$\begin{aligned}
& \widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C)|z) && \text{(H14)}\\
= \ & \widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C)|\{mpk_i\}_{i\in[l]}, C, \{ct_i\}_{i\in[q_e]}, l_L\text{-}leak, \{ct_i^*\}_{i\in[q_{lr}]}, \mathsf{Leak}(\{sk_f\}_{f\in\{f\}})) && \text{(H15)}\\
\geqslant \ & \widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C)|\{mpk_i\}_{i\in[l]}, C, \{ct_i^*\}_{i\in[q_{lr}]}, \mathsf{Leak}(\{sk_f\}_{f\in\{f\}})) - l_L && \text{(H16)}\\
\geqslant \ & \widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C)|\{mpk_i\}_{i\in[l]}, C, \mathsf{Leak}(\{sk_f\}_{f\in\{f\}})) - l_L - q_{lr}.l_{\mathsf{LAF}} - q_{lr}.m && \text{(H17)}
\end{aligned}$$

$$= \widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*,C)|\{mpk_i\}_{i\in[l]},C,td^*) - l_L - q_{lr}.l_{\mathsf{LAF}} - q_{lr}.m \quad \text{(H18)}$$

$$= \widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*,C)|mpk^*,C,td^*) - l_L - q_{lr}.l_{\mathsf{LAF}} - q_{lr}.m \quad \text{(H19)}$$

$$\geqslant \nu - l_L - q_{lr}.l_{\mathsf{LAF}} - q_{lr}.m \quad \text{(H20)}$$

Since $\{ct_i\}_{i\in[q_e]}$ are computed by the ENC oracle under the master public key $\{mpk_i\}_{i\in[l]}$, the leaked information about the master secret key $msk^* = sk^*$ from these ciphertexts has been implied in $\{mpk_i\}_{i\in[l]}$, then by Lemma **??**, the Eq. H16 holds. In the challenge ciphertext $\{ct_i^*\}_{i\in[q_{lr}]}$, only $\{U_i^*\}_{i\in[q_{lr}]}$ and $\{\pi_i^*\}_{i\in[q_{lr}]}$ will leak the information about the master secret key and since U_i^* and π_i^* have respectively at most 2^m and $2^{l_{\mathsf{LAF}}}$ possible values, therefore, Eq. H17 holds. Moreover, as all randomness used for generating the private keys $\{sk_f\}_{f\in\{f\}}$ queried by the adversary are independent of sk^*, hence, only the trapdoor td^* may leak the information of sk^*. So Eq. H18 holds. Again, since each $sk_i \in \{sk_i\}_{i\in[l]}$ is chosen uniformly and dependently from its space $\mathcal{SK}_i$, hence Eq. H19 holds. The last equation H20 follows from the ϵ_1-*universal* property of the THPS. In fact, the event $\overline{T}$ happening means that the tag $t = ((C,s,U),t_c)$ is an injective tag which maintains the entropy of the injective function $\mathsf{LAF}_{lpk^*,t}(\cdot)$. In addition, since the probability that the adversary can find a bad ciphertext is at most $2^{l_L+q_{lr}.l_{\mathsf{LAF}}+q_{lr}.m}/2^\nu$, hence, when this ciphertext is not bad, the adversary can eliminate one encapsulated key K from the space 2^ν. Thus we have

$$\Pr[\mathsf{bad}_{ct}|\overline{T}] \leqslant q_d.2^{l_L+q_{lr}.l_{\mathsf{LAF}}+q_{lr}.m}/(2^\nu - q_{bad}). \quad \text{(H21)}$$

Combining Eqs. H13 and H21, we get

$$\Pr[\mathsf{bad}_{ct}] \leqslant q_d.2^{l_L+q_{lr}.l_{\mathsf{LAF}}+q_{lr}.m}/(2^\nu - q_{bad}) + q_{bad}.\mathsf{Adv}^{\mathsf{evs}}_{\mathsf{LAF},\mathcal{A}_{evs}}(\lambda), \quad \text{(H22)}$$

which is negligible in λ. As a result, the probability that the adversary $\mathcal{A}'$ outputs a ciphertext $ct = (C,s,U,\pi,t_c)$ with $C \in \mathcal{C}\backslash\mathcal{V}$ that makes $\mathcal{G}_f(ct) \neq \mathsf{S}.\mathcal{G}_f(ct)$ is negligible, which implies that the circuit family $(\mathcal{G}_f, \mathsf{S}.\mathcal{G}_f, z, \mathsf{Sam})$ is a differing-inputs distribution. Subsequently, we use the the distinguishable advantage between $\mathsf{DIO}(\mathcal{G}_f)$ and $\mathsf{DIO}(\mathsf{S}.\mathcal{G}_f)$ to bound the difference between games G_5 and G_6. Assume that $\mathcal{A}_2$ makes a total of q_{sk} KeyGen queries. We define hybrids $\mathsf{G}_{5,i}$, $0 \leqslant i \leqslant q_{sk}$, in which we respond to the first $q_{sk} - i$ queries as in G_5, and respond to the last i queries as in G_6. Let $\mathcal{B}$ be the DIO challenger, then the adversary $\mathcal{A}_{dio}$ is constructed as follows:

1. $\mathcal{A}_{dio}$ first honestly generates $(\{mpk_{i'}\}_{i'\in[l]}, \{msk_{i'}\}_{i'\in[l]}, \{td_{i'}\}_{i'\in[l]}, st')$.
2. For the first $(q_{sk}-i-1)$ key queries, $\mathcal{A}_{dio}$ computes the key for f as in G_5 and for the last i key queries, $\mathcal{A}_{dio}$ computes the key for f as in G_6.
3. For the $(q_{sk}-i)'$th key query from $\mathcal{A}_2$, $\mathcal{A}_{dio}$ chooses a PRF key $r' \leftarrow_\$ \mathcal{R}_\lambda$, computes $r'_{\mathsf{P}} = \mathsf{PF.Punc}(r', \{ct_j^*\}_{j\in[q_{lr}]})$ and $\{y_j^*\}_{j\in[q_{lr}]} = \{f(x_j; \mathsf{PF.Eval}(r', ct_j^*))\}_{j\in[q_{lr}]}$. It then uses mpk^*, msk^*, r' and f to construct the circuit $\mathcal{G}_f$ and uses mpk^*, td^*, r'_{P}, $\{ct_j^*\}_{j\in[q_{lr}]}$ and $\{y_j^*\}_{j\in[q_{lr}]}$ to build the circuit $\mathsf{S}.\mathcal{G}_f$. Next it sends the two circuits to $\mathcal{B}$ and receives $sk_f = \mathsf{DIO}(cir)$ from $\mathcal{B}$, where cir denotes either $\mathcal{G}_f$ or $\mathsf{S}.\mathcal{G}_f$. Finally, sk_f is sent to the adversary $\mathcal{A}_2$. Note that in this phase, all these simulations can be done since, according to the security definition, the values $\{ct_j^*\}_{j\in[q_{lr}]}$ and the target key have been generated priori to the KeyGen oracle queries.
4. $\mathcal{A}_{dio}$ simulates the rest parts of the experiment for $\mathcal{A}_2$ in the same manner as in G_5 and G_6.
5. At the end, $\mathcal{A}_{dio}$ sends the output of the experiment to $\mathcal{A}_2$ and returns its results to $\mathcal{B}$.

Now if $\mathcal{B}$ returns the obfuscation of $\mathcal{G}_f$, then $\mathcal{A}_{dio}$ simulates game $\mathsf{G}_{5,i}$ for $\mathcal{A}$, else it simulates game $\mathsf{G}_{5,i+1}$ for $\mathcal{A}$. Thus, by a hybrid argument, we have

$$\Pr[\mathsf{G}_5|\overline{\mathsf{coll}}] - \Pr[\mathsf{G}_6|\overline{\mathsf{coll}}] \leqslant q_{sk}.\mathsf{Adv}^{\mathsf{dio}}_{\mathcal{G}_f,\mathsf{S}.\mathcal{G}_f,\mathcal{A}_{dio}}(\lambda).$$

Proof of Claim 7 . Assume that $\mathcal{A}_2$ makes a total of q_{sk} key queries. We consider q_{sk} intermediate hybrids $\mathsf{G}_{6,i}$ for $0 \leqslant i \leqslant q_{sk}$ in which we respond to the first $q_{sk} - i$ key queries as in G_6, and the remaining i key queries as in G_7. We show that if there exists a PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that can distinguish hybrid $\mathsf{G}_{6,i}$ and hybrid $\mathsf{G}_{6,i+1}$ with non-negligible probability, then we can construct a PPT adversary $\mathcal{A}'_{prf}$ which can break the security of puncturable PRF PF with non-negligible probability. Let $\mathcal{B}$ be the puncturable PRF challenger, then the construction of $\mathcal{A}'_{prf}$ is as follows.

1. The adversary $\mathcal{A}'_{prf}$ first honestly generates $(\{mpk_{i'}\}_{i'\in[l]}, \{msk_{i'}\}_{i'\in[l]}, \{td_{i'}\}_{i'\in[l]}, st')$.
2. For the first $q_{sk} - i - 1$ key queries from $\mathcal{A}_2$, $\mathcal{A}'_{prf}$ responds in the same manner as in game G_6. For the last i key queries, $\mathcal{A}'_{prf}$ responds as in game G_7.
3. For the $(q_{sk}-i)'$th key query, $\mathcal{A}'_{prf}$ first sends $\{ct_j^*\}_{j\in[q_{lr}]}$ to its challenger $\mathcal{B}$ and receives the pair $(r'_{\mathsf{P}}, \{r_j\}_{j\in[q_{lr}]})$, where $r'_{\mathsf{P}} = \mathsf{PF.Punc}(r', \{ct_j^*\}_{j\in[q_{lr}]})$, while r_j is either $\mathsf{PF.Eval}(r', ct_j^*)$ or a uniform and random value in $\mathcal{R}_\lambda$. The adversary $\mathcal{A}'_{prf}$ then computes $y_j^* = f(x_j; r_j)$ for all $j \in [q_{lr}]$ and defines the circuit $\mathsf{S}.\mathcal{G}_f$ using the values mpk^*, td^*, r'_{P}, $\{ct_j^*\}_{j\in[q_{lr}]}$, $\{y_j^*\}_{j\in[q_{lr}]}$ and f. Finally, $\mathcal{A}'_{prf}$ sets $sk_f = \mathsf{DIO}(\mathsf{S}.\mathcal{G}_f)$ and sends sk_f to the adversary $\mathcal{A}_2$. Note that all these simulations can be done since the values $\{ct_j^*\}_{j\in[q_{lr}]}$ and target key have been generated priori to the KeyGen oracle queries, so the values $\{y_j^*\}_{j\in[q_{lr}]}$ and sk_f can be computed by $\mathcal{A}'_{prf}$.
4. $\mathcal{A}'_{prf}$ simulates the rest of the experiment in the same manner as in game G_6 and game G_7.
5. At the end, the adversary $\mathcal{A}'_{prf}$ sends the outputs of the experiment to $\mathcal{A}_2$ and returns its results to $\mathcal{B}$.

Therefore, if $\mathcal{B}$ returns $r_j = \mathsf{PF.Eval}(r', ct_j^*)$, then $\mathcal{A}'_{prf}$ perfectly simulates game $\mathsf{G}_{6,i}$ for $\mathcal{A}_2$, else it simulates game $\mathsf{G}_{6,i+1}$ for $\mathcal{A}_2$. Thus, by a hybrid argument, we have

$$\Pr[\mathsf{G}_6|\overline{\mathsf{coll}}] - \Pr[\mathsf{G}_7|\overline{\mathsf{coll}}] \leqslant q_{sk}.\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PF},\mathcal{A}'_{prf}}(\lambda).$$

Proof of Claim 8 . Let Event_C denote the event that a ciphertext such that $C \in \mathcal{C}\backslash\mathcal{V}$ is rejected in game G_8 while is not rejected in game G_7. Then G_7 and G_8 behave identical until the event Event_C arises. Obviously, the advantage that an adversary distinguishes game G_7 from G_8 can be bounded by the probability that the event Event_C happens. Thus we have

$$\Pr[\mathsf{G}_7|\overline{\mathsf{coll}}] - \Pr[\mathsf{G}_8|\overline{\mathsf{coll}}] \leqslant \Pr[\mathsf{Event}_C].$$

Now we analyze the upper bound of the probability $\Pr[\mathsf{Event}_C]$. If we call a tag t such that $t = t^*$ a repeated tag, then by T we denote the event that in game G_7, there exists a decryption query $ct = (C, s, U, \pi, t_c)$ with $t = ((C, s, U), t_c)$ being a non-injective, non-repeated tag. Then we have

$$\Pr[\mathsf{Event}_C] = \Pr[\mathsf{Event}_C \wedge \overline{T}] + \Pr[\mathsf{Event}_C \wedge T] \leqslant \Pr[\mathsf{Event}_C|\overline{T}] + \Pr[T].$$

First we claim that if there exists an adversary $\mathcal{A}_{evs}$ that can break the evasiveness security of the lossy algebraic filter LAF with advantage $\mathsf{Adv}^{\mathsf{evs}}_{\mathsf{LAF},\mathcal{A}_{evs}}(\lambda)$, then the probability $\Pr[T]$ can be upper bounded by this advantage. We build the algorithm $\mathcal{A}_{evs}$ as follows. Assume $\mathcal{A}_2$ makes a total of q_d decryption queries and let $\mathcal{B}$ denote the LAF challenger.

1. The adversary $\mathcal{A}_{evs}$ first honestly generates $(\{mpk_i\}_{i\in[l]}, \{msk_i\}_{i\in[l]}, \{td_i\}_{i\in[l]}, st')$ which are generated in the same way as game G_7 except that it receives $\{lpk_i\}_{i\in[l]}$ from his LAF challenger $\mathcal{B}$.
2. For the LR queries from $\mathcal{A}_2$, the adversary $\mathcal{A}_{evs}$ first constructs t_a^*, and then delivers t_a^* to its LAF challenger $\mathcal{B}$ and receives t_c^*. Next, $\mathcal{A}_{evs}$ constructs the other parts of the challenge ciphertext ct^* for $\mathcal{A}_2$ as in game G_7.
3. For the DEC queries from $\mathcal{A}_2$, since the adversary $\mathcal{A}_{evs}$ has $msk^* = sk^*$, he can perfectly simulates the decryption queries for $\mathcal{A}_2$.
4. $\mathcal{A}_{evs}$ simulates the rest of the experiment in the same manner as in G_7 and G_8.
5. Finally, the adversary $\mathcal{A}_{evs}$ sends the output of the experiment to $\mathcal{A}_2$ and returns its results to $\mathcal{B}$.

At the end of the simulation, $\mathcal{A}_{evs}$ chooses $j \in [q_d]$ uniformly and outputs the tag $t = ((C, s, U), t_c)$ extracted from $\mathcal{A}_2$'s j-th decryption query (C, s, U, π, t_c). Obviously, if the event T happens, $t = ((C, s, U), t_c)$ is a non-injective tag with probability at least $1/q_d$, namely

$$\Pr[T] \leqslant q_d.\mathsf{Adv}^{\mathsf{evs}}_{\mathsf{LAF},\mathcal{A}_{evs}}(\lambda). \tag{H23}$$

Next, we claim that the probability $\Pr[\mathsf{Event}_C|\overline{T}]$ can be upper bounded by

$$q_d.2^{l_L + q_{lr}.l_{\mathsf{LAF}} + q_{lr}.m}/(2^\nu - q_d).$$

Without loss of generality, assume that $ct = (C, s, U, \pi, t_c)$ is the first ciphertext that makes Event_C happen conditioned on $\overline{T}$. Namely, both $C \in \mathcal{C}\backslash\mathcal{V}$ and $\pi = \mathsf{LAF}_{lpk^*,t}(\mathsf{THPS.Priv}(sk^*, C))$ hold and $t = ((C, s, U), t_c)$ is an injective tag. Since the values $\{mpk_i\}_{i\in[l]}$, C, $\{ct_i\}_{i\in[q_e]}$, l_L-*leak*, $\{ct_i^*\}_{i\in[q_{lr}]}$ and $\mathsf{Leak}(\{sk_f\}_{f\in\{f\}})$ are in $\mathcal{A}$'s view, let $V_\mathcal{A}$ denote $\mathcal{A}$'s view, then according to the analysis of equation H14, we have

$$\begin{aligned}
&\widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C)|V_\mathcal{A}) && \text{(H24)}\\
= {}&\widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C)|\{mpk_i\}_{i\in[l]}, C, \{ct_i\}_{i\in[q_e]}, l_L\text{-}leak, \{ct_i^*\}_{i\in[q_{lr}]}, \mathsf{Leak}(\{sk_f\}_{f\in\{f\}})) && \text{(H25)}\\
\geqslant {}&\widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C)|\{mpk_i\}_{i\in[l]}, C, \{ct_i^*\}_{i\in[q_{lr}]}, \mathsf{Leak}(\{sk_f\}_{f\in\{f\}})) - l_L && \text{(H26)}\\
\geqslant {}&\widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C)|\{mpk_i\}_{i\in[l]}, C, \mathsf{Leak}(\{sk_f\}_{f\in\{f\}})) - l_L - q_{lr}.l_{\mathsf{LAF}} - q_{lr}.m && \text{(H27)}\\
= {}&\widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C)|\{mpk_i\}_{i\in[l]}, C, td^*) - l_L - q_{lr}.l_{\mathsf{LAF}} - q_{lr}.m && \text{(H28)}\\
= {}&\widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C)|mpk^*, C, td^*) - l_L - q_{lr}.l_{\mathsf{LAF}} - q_{lr}.m && \text{(H29)}\\
\geqslant {}&\nu - l_L - q_{lr}.l_{\mathsf{LAF}} - q_{lr}.m && \text{(H30)}
\end{aligned}$$

Similar to Claim 6, the event $\overline{T}$ happening means that the tag $t = ((C, s, U), t_c)$ is an injective tag which maintains the entropy of the injective function $\mathsf{LAF}_{lpk^*,t}(\cdot)$. In addition, since in game G_7 the decryption algorithm accepts such an invalid ciphertext with probability at most $2^{l_L + q_{lr}.l_{\mathsf{LAF}} + q_{lr}.m}/2^\nu$, when such a ciphertext is rejected, the adversary can eliminate one encapsulated key K from the space 2^ν. Thus we have

$$\Pr[\mathsf{Event}_C|\overline{T}] \leqslant q_d.2^{l_L + q_{lr}.l_{\mathsf{LAF}} + q_{lr}.m}/(2^\nu - q_d). \tag{H31}$$

Combining Eqs. H23 and H31, we get Eq. H10.

Proof of Claim 9 . For each $i \in [q_{lr}]$, conditioned on $\mathcal{A}$'s view

$$V'_\mathcal{A} = (\{mpk_j\}_{j\in[l]}, \{ct_i\}_{i\in[q_e]}, C_i^*, \{C_j^*\}_{i\neq j\in[q_{lr}]}, \{\pi_j\}_{j\in[q_{lr}]}, \{U_j^*\}_{i\neq j\in[q_{lr}]}, l_L\text{-}leak, \mathsf{Leak}(\{sk_f\}_{f\in\{f\}})),$$

we can get the lower bound of $\widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C_i^*)|V'_\mathcal{A})$ as below.

$$\begin{aligned}
&\widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C_i^*)|V'_\mathcal{A}) && \text{(H32)}\\
\geqslant {}&\widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C_i^*)|\{mpk_j\}_{j\in[l]}, \{C_j^*\}_{j\in[q_{lr}]}, \{U_j^*\}_{i\neq j\in[q_{lr}]}, \mathsf{Leak}(\{sk_f\}_{f\in\{f\}})) - q_{lr}.l_{\mathsf{LAF}} - l_L && \text{(H33)}\\
\geqslant {}&\widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C_i^*)|\{mpk_j\}_{j\in[l]}, \{C_j^*\}_{j\in[q_{lr}]}, \{U_j^*\}_{i\neq j\in[q_{lr}]}, td^*) - q_{lr}.l_{\mathsf{LAF}} - l_L && \text{(H34)}\\
\geqslant {}&\widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C_i^*)|\{mpk_j\}_{j\in[l]}, \{C_j^*\}_{j\in[q_{lr}]}, td^*) - (q_{lr} - 1).m - q_{lr}.l_{\mathsf{LAF}} - l_L && \text{(H35)}\\
\geqslant {}&\widetilde{\mathsf{H}}_\infty(\mathsf{THPS.Priv}(sk^*, C_i^*)|\{mpk_j\}_{j\in[l]}, C_i^*, td^*) - (q_{lr} - 1).m - q_{lr}.l_{\mathsf{LAF}} - l_L && \text{(H36)}\\
\geqslant {}&\nu - (q_{lr} - 1).m - q_{lr}.l_{\mathsf{LAF}} - l_L && \text{(H37)}
\end{aligned}$$

Obviously, Eqs. H33,H34,H35 hold. Since for each $i \in [q_{lr}]$, C_i^* is chosen uniformly and randomly from $\mathcal{C}\backslash\mathcal{V}$ and independent of sk^*, thereby, Eq. H36 holds. The last equation H37 follows from the ϵ_1-*universal* property of the THPS.

We define an intermediate game G_8' which is the same as G_8 except that U_i^* in each challenge ciphertext in the set $\{ct_i^*\}_{i\in[q_{lr}]}$ is chosen uniformly and randomly from $\{0,1\}^m$. In the following, we first show that game G_8 is indistinguishable from game G_8', then in turn give proofs that G_8' and G_9 are indistinguishable. We consider q_{lr} intermediate hybrids $\mathsf{G}_{8,i}$ for $0 \leqslant i \leqslant q_{lr}$ where in $\mathsf{G}_{8,i}$, we respond to the first $q_{lr}-i$ LR queries as in game G_8 and the remaining i LR queries as in G_8'.

Taking $\mathsf{THPS.Priv}(sk^*, C_i^*)$ as an input to the average-case $((\nu-(q_{lr}-1).m-q_{lr}.l_{\mathsf{LAF}}-l_L),\epsilon_2)$-strong extractor, we have that $\mathsf{Ext}(\mathsf{THPS.Priv}(sk^*, C_i^*), s_i^*)$ is ϵ_2-close to uniform distribution given $\mathcal{A}$'s view, thus we have $\mathsf{Pr}[\mathsf{G}_{8,i}]-\mathsf{Pr}[\mathsf{G}_{8,i+1}] \leqslant \epsilon_2$ which leads to $\mathsf{Pr}[\mathsf{G}_8]-\mathsf{Pr}[\mathsf{G}_{8'}] \leqslant q_{lr}.\epsilon_2$. Likewise, we can get $\mathsf{Pr}[\mathsf{G}_8']-\mathsf{Pr}[\mathsf{G}_9] \leqslant q_{lr}.\epsilon_2$. In addition, since for each $i \in [q_{lr}]$, the security definition requires that the equation $f(x_i;r)=f(0;r)$ should hold for the information-theoretically fixed r by x_i, therefore, sk_f do not help the adversary distinguish games G_8 and G_9. Hence, we have

$$\mathsf{Pr}[\mathsf{G}_8|\overline{\mathsf{coll}}]-\mathsf{Pr}[\mathsf{G}_9|\overline{\mathsf{coll}}] \leqslant 2.q_{lr}.\epsilon_2.$$

Proof of Claim 10 . Assume that $\mathcal{A}_2$ makes a total of q_d DEC queries. We consider q_d intermediate hybrids $\mathsf{G}_{9,i}$ for $0 \leqslant i \leqslant q_d$ in which we respond to the first q_d-i decryption queries as in game G_9, and the remaining i decryption queries as in G_{10}. We show that if there exists a PPT adversary $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$ that can distinguish $\mathsf{G}_{9,i}$ and $\mathsf{G}_{9,i+1}$, then there exists a PPT adversary $\mathcal{A}_{prf}$ that can break the security of the puncturable PRF PF. Let $\mathcal{B}$ denote the PRF challenger, then the reduction is as follows.

1. The adversary $\mathcal{A}_{prf}$ first honestly generates $(\{mpk_{i'}\}_{i'\in[l]}, \{msk_{i'}\}_{i'\in[l]}, \{td_{i'}\}_{i'\in[l]}, st')$.
2. For the first q_d-i-1 DEC queries from $\mathcal{A}_2$, $\mathcal{A}_{prf}$ responds in the same manner as in game G_9. For the last i DEC queries, $\mathcal{A}_{prf}$ responds in the same manner as in game G_{10}.
3. For the $(q_d-i)'$th DEC query such as $(ct,g,1,n)$, if $ct \notin \{ct_i^*\}_{i\in[q_{lr}]}$, $\mathcal{A}_{prf}$ first gets x by decrypting ct with the master secret key $msk^*=sk^*$ and then submits ct to the PRF challenger $\mathcal{B}$ and receives r'' which is either computed as $r''=\mathsf{PF.Eval}(r',ct)$ or chosen uniformly and randomly. Finally the adversary $\mathcal{A}_{prf}$ computes $y=g(x;r'')$ and sends it to the adversary $\mathcal{A}_2$. If $ct \in \{ct_i^*\}_{i\in[q_{lr}]}$, $\mathcal{A}_{prf}$ outputs $\perp$ and stops.
4. $\mathcal{A}_{prf}$ simulates the rest of the experiment in the same manner as in G_9 and G_{10}.
5. At the end, $\mathcal{A}_{prf}$ sends the output of the experiment to $\mathcal{A}_2$ and returns its results to $\mathcal{B}$.

If $\mathcal{B}$ returns $r''=\mathsf{PF.Eval}(r',ct)$, then $\mathcal{A}_{prf}$ perfectly simulates game $\mathsf{G}_{9,i}$ for $\mathcal{A}_2$, else it simulates game $\mathsf{G}_{9,i+1}$ for $\mathcal{A}_2$. Where r' is a random PRF key chosen by the challenger $\mathcal{B}$. Thus, by a hybrid argument, we have

$$\Pr[\mathsf{G}_9|\overline{\mathsf{coll}}]-\Pr[\mathsf{G}_{10}|\overline{\mathsf{coll}}] \leqslant q_d.\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PF},\mathcal{A}_{prf}}(\lambda).$$

References

1 M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *Advance in EUROCRYPT 2006*, pages 409–426, 2006.

2 R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In L. R. Knudsen, editor, *Advance in EUROCRYPT 2002*, volume 2332, pages 45–64. Springer, Heidelberg, 2002.

3 V. Goyal, A. Jain, V. Koppula, and A. Sahai. Functional encryption for randomized functionalities. In Y. Dodis and J.B. Nielsen, editors, *TCC 2015, Part II, LNCS 9015*, pages 325–351, 2015.

4 D. Hofheinz. Circular chosen-ciphertext security with compact ciphertexts. In T. Johansson and P. Q. Nguyen, editors, *Advances in EUROCRYPT 2013*, volume 7881, pages 520–536, 2013.

5 K. G. Paterson, J. C. N. Schuldt, and D. L. Sibborn. Related randomness attacks for public key encryption. In H. Krawczyk, editor, *PKC 2014*, volume LNCS8383, pages 465–482. Springer 2014, 3 2014.

6 B.D. Qin and S.L. Liu. Leakage-resilient chosen-ciphertext secure public-key encryption from hash proof system and one-time lossy filter. In K. Sako and P. Sarkar, editors, *Advance in ASIACRYPT 2013*, pages 381–400, 2013.

7 H. Shacham T. Ristenpart and T. Shrimpton. Careful with composition: Limitations of the indiferentiability framework. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of LNCS, pages 487–506. Springer, May 2011.