Generalized (Identity-Based) Hash Proof System and Its Applications

Yu Chen[1], Zongyang Zhang[2], Dongdai Lin[1], Zhenfu Cao[2]

[1]State Key Laboratory of Information Security (SKLOIS),
Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100095, China
{chenyu,ddlin}@iie.ac.cn
[2]Department of Computer Science and Engineering,
Shanghai Jiao Tong University, China
{zongyangzhang,zfcao}@sjtu.edu.cn

Abstract. In this work, we generalize the paradigm of hash proof system (HPS) proposed by Cramer and Shoup [CS02]. In the central of our generalization, we lift subset membership problem to distribution distinguish problem. Our generalized HPS clarifies and encompass all the known public-key encryption (PKE) schemes that essentially implement the idea of hash proof system. Moreover, besides existing smoothness property, we introduce an additional property named anonymity for HPS. As a natural application, we consider anonymity for PKE in the presence of key-leakage, and provide a generic construction of leakage-resilient anonymous PKE from anonymous HPS. We then extend our generalization to the identity-based setting. Concretely, we generalize the paradigm of identity-based hash proof system (IB-HPS) proposed by Boneh *et al.* [BGH07] and Alwen *et al.* [ADN$^+$10], and introduce anonymity for it. As an interesting application of anonymous IB-HPS, we consider security for public-key encryption with keyword search (PEKS) in the presence of token-leakage, and provide a generic construction of leakage-resilient secure PEKS from leakage-resilient anonymous IBE, which in turn is based on anonymous IB-HPS.

Key words: (identity-based) hash proof system, leakage-resilience, anonymity, public-key encryption with keyword search

1 Introduction

In EUROCRYPT 2002, Cramer and Shoup [CS02] abstracted their earlier public-key encryption (PKE) scheme [CS98] to the paradigm of hash proof system (HPS) with the initial purpose to provide a framework for the construction of CCA-secure PKE. Thereafter, HPS and its variants have found numerous applications beyond CCA security, including password-based authenticated key exchange (PAKE) [GL06, KV09], oblivious transfer [HK12], extractable commitment [ACP09], privacy-preserving interactive protocols [BPV12], leakage-resilient PKE [NS09, ADN$^+$10], threshold cryptosystems [LY12], lossy encryption [BHY09] and thus selective opening secure PKE [HLOV11], lossy trapdoor hash functions (LTDF) [PW08, Wee12, HO12], and thus deterministic PKE [BBO07, BFOR08, BS11].

Briefly, HPS consists of two ingredients, namely subset membership problem (SMP) and projective hash family (PHF). The SMP defines a set X and a language $L \subset X$, from which a member x can be efficiently sampled with a witness w. Intuitively, a subset membership problem is hard if it is computationally impossible to distinguish random members $x \in L$ from random non-members $x \in X \backslash L$. The PHF with projection $\alpha : SK \rightarrow PK$ is a family of hash functions H indexed by SK with domain X. HPS connects SMP and PHF by providing two algorithms to evaluate H_{sk} on L, that is one can either evaluate $\mathsf{H}_{sk}(x)$ privately using sk or publicly using the witness w for x. Basically, we require HPS to be projective and smooth. The projective property stipulates that the action of H_{sk} on L is determined by $\alpha(sk)$, while the smooth property stipulates that the action of H_{sk} on $X \backslash L$ is undetermined. In applications of HPS, the projective property is usually used to guarantee correctness while the hardness of subset membership problem and smooth property are used to establish security. We further explain this idea by taking the construction CPA-secure PKE from smooth HPS as a concrete example. The construction of PKE from HPS is immediate: to encrypt message m under public key pk, one samples an instance x from L as the ciphertext, then computes the hash value $y \leftarrow \mathsf{H}_{sk}(x)$ publicly using the witness w, and finally masks m with y. The correctness follows from the projection property, which ensures the recipient can recover y by computing H_{sk} privately with sk. The security follows from the smoothness property and the hardness of subset membership problem, that is, when sampling x from $X \backslash L$ the corresponding hashing value y is statistically close to uniform and no PPT adversary can notice this switch (from $x \leftarrow L$ to $x \leftarrow X \backslash L$).

1.1 Related Works

Gennaro and Lindell [GL06] tailored HPS by defining $L = \{(c, m)\}$, where c is a non-malleable commitment to m, then used this variant to build a framework of PAKE. Abdalla *et al.* [ACP09] modified HPS to support disjunctions and conjunctions of language, then constructed extractable commitment from it. Halevi and Kailai [HK12] presented a construction of two-message oblivious transfer from smooth HPS with verifiable smoothness. Hemenway *et al.* [HLOV11] formally defined homomorphic smooth HPS, and showed how to construct lossy encryption and thus selective opening secure PKE from it. Hemenway and Ostrovsky [HO12] showed the usage of homomorphic smooth HPS in the construction of LTDF. Libert and Yung [LY12] extended HPS to all-but-one perfectly sound threshold HPS, then used it to construct non-interactive CCA-secure PKE. Wee [Wee12] proposed the notion of dual HPS, in which H is viewed in a dual way (indexed by X with domain SK), and invertibility on $X \backslash L$ is required instead of smoothness. As shown by [Wee12], dual HPS immediately yields elegant and simple constructions of LTDF and deterministic PKE.

Boneh *et al.* [BGH07] extended HPS to the identity-based setting, which they called HPS with trapdoor. Alwen *et al.* [ADN$^+$10] then formally define the notion of identity-based hash proof system (IB-HPS) in the description of identity-based key encapsulation mechanism.

1.2 Motivation

In this paper, we focus on the original (IB)-HPS and its application in encryption schemes. While (IB)-HPS can be constructed for languages related to Diffie-Hellman like assumptions (such as DDH, DLIN, and DBDH) and number-theoretic assumptions (such as QR and DCR), it is interesting to realize the difficulty of construct efficient (IB)-HPS for lattice-based assumptions (such as LWE)[1]. The main reason is that the original paradigm of (IB)-HPS insists that L and $X\backslash L$ are disjointed. This requirement works well with the concept of valid ciphertext and invalid ciphertext, and the original paradigm can be interpreted exactly by languages related to DH-like and number-theoretic assumptions (those with precise algebraic property). However, such requirement rules out lattice-based instantiations. We first note that more formally, subset membership problem should be reformulated as distribution distinguish problem, which states that a uniform distribution on L is computationally indistinguishable from a uniform distribution on $X\backslash L$. We then observe that for most usages of HPS, it is the indistinguishability between two modes (one is for real system and the other is for simulation) that really matters. Hence, the supports of the two distributions are not necessarily disjointed, and the uniform requirement is also not necessary. Moreover, in the original paradigm of HPS the language L is independent of PK, which means that ciphertext is unrelated to the public key. To encompass a broad class of encryption schemes that relying on HPS, L should be viewed as a collection of languages indexed by PK.

The classical security definitions for encryption schemes are mainly concerned with data privacy, that a ciphertext does not reveal the data it encrypts. The well-known notions such as one-wayness (OW) and indistinguishability (IND) are both directed at capturing different levels of data privacy. However, many newly emerged applications such as cloud computing also require key privacy, that is ciphertext does not reveal the key under which it encrypts. The concept of key privacy (or anonymity) was first formalized in the context of symmetric-key encryption [AR02, Des00, Fis99] and was later extended to the context of public-key encryption (PKE) [BBDP01] and identity-based encryption (IBE) [ABC$^+$05]. It is easy to see that the goals of data privacy and key privacy are orthogonal [BBDP01, ZHI07]. There exist encryption schemes which satisfy the strongest data privacy but still lack the weakest key privacy. For instance, given a PKE scheme with certain level of data privacy, we can construct a variant of it by simply adding the public key to the ciphertexts. The new PKE scheme preserves original data privacy but obviously loses all the key privacy. Similar examples also exists in the identity-based settings, where we just make the identity explicit in the ciphertexts.

Unlike that data privacy has already been extensively studied in the literature, key privacy is comparatively less studied. Recently, researchers start to consider data privacy in the presence of key-leakage attacks, as in real life an adversary might gain some partial information about secret keys, by observing behavior of the protocol executions, or measuring the places where they are stored, via cold-boot attacks, electromagnetic measurements and many more. A large body of work [NS09, ADN$^+$10, CDRW10, LRW11, CLC11] have already emerged on constructing encryption schemes with data privacy against various key-leakage attacks. However, all these work only focus on classical security (data privacy) but not anonymity (key privacy). On one hand, for encryption schemes derived from (IB)-HPS, while the smooth property of (IB)-HPS implies data privacy, there is no property of (IB)-HPS ready to imply anonymity. On the other

[1] We note two exceptions here. Katz and Vaikuntanathan [KV09] presented a variant of HPS for the application in PAKE and a gave a latticed-based instantiation. However, their variant of HPS differs much from the original HPS in the definition of language and projection. Alwen *et al.* [ADN$^+$10] gave a lattice-based IB-HPS in [ADN$^+$10]. However, the construction can not fit their definition of IB-HPS exactly as the valid and invalid ciphertext space are identical, i.e., what the so called invalid encapsulation algorithm output may also be a valid ciphertext.

hand, we note that (IB)-HPS is one of the two known methodologies[2] to achieve leakage-resilience (mainly benefit from the redundancy introduced to the secret key). Hence, we are motivated to seek an additional property for (IB)-HPS which implies anonymity for the resulting encryption schemes, and consider the construction of leakage-resilient anonymous encryption schemes via IB-HPS with such property.

1.3 Our Contribution

- Generalize the paradigm of HPS and present a latticed-based instantiation.
- Introduce a new property named anonymity for HPS; formally define the anonymity for PKE in the presence of key-leakage.
- Show how to convert smooth and anonymous HPS to leakage-resilient one, and further show how to construct leakage-resilient secure and anonymous PKE.
- Generalize the paradigm of identity-based HPS and present six instantiations based on the ideas behind prior IBE schemes.
- Introduce a new property named anonymity for IB-HPS; formally define the anonymity for IBE in the presence of key-leakage.
- Show how to convert smooth and anonymous IB-HPS to leakage-resilient one, and further show how to construct leakage-resilient secure and anonymous IBE.
- Introduce the notion of leakage-resilient security for public-key encryption with keyword search (PEKS), and construct leakage-resilient secure PEKS via leakage-resilient anonymous IBE, which is in turn derived from anonymous IB-HPS.

2 Definitions and Preliminaries

2.1 Notation

For a finite set S, we denote by $x \xleftarrow{R} S$ the action of uniformly choosing an element x from S, and by $x, y, z \xleftarrow{R} S$ the action of independently and uniformly choosing all x, y, z from S. We denote by U_S the uniform distribution over S. For an integer $n \in \mathbb{N}$, we denote by U_n the uniform distribution over $\{0,1\}^n$. For a distribution D, we denote by $|D|$ its support, and by $x \leftarrow D$ the action of choosing x according to D. Throughout this paper, $\kappa \in \mathbb{N}$ denotes the security parameter. We denote by $\mathsf{negl}(\kappa)$ an unspecified function negligible in κ. We use PPT to denote probabilistic polynomial-time. If $\mathcal{A}$ is a randomized algorithm, we write $z \leftarrow \mathcal{A}(x_1, \ldots, x_n; r)$ to indicate that $\mathcal{A}$ outputs z on inputs $(x_1, \ldots, x_n)$ and random coins r. Sometimes for brevity, we omit r and write $z \leftarrow \mathcal{A}(x_1, \ldots, x_n)$ when it is not necessary to make explicit the random coins $\mathcal{A}$ uses.

2.2 Assumptions

Let $\mathsf{BLGroupGen}$ be a PPT algorithm that takes as input a security parameter κ and output a tuple $(p, \mathbb{G}, \mathbb{G}_T, e)$, where p is a κ-bit prime, $\mathbb{G}$ and $\mathbb{G}_T$ are two groups of order p, e is a bilinear map from $\mathbb{G} \times \mathbb{G}$ to $\mathbb{G}_T$. The following three assumptions are all defined with respect to parameters $(p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathsf{BLGroupGen}(\kappa)$.

Decisional Bilinear Diffie-Hellman Assumption. The decisional bilinear Diffie-Hellman (DBDH) assumption [BF03] is that the distributions $(g, g^a, g^b, g^c, e(g,g)^{abc})$ and (g, g^a, g^b, g^c, Z) are computationally indistinguishable, where $g \xleftarrow{R} \mathbb{G}^*$, $a, b, c \xleftarrow{R} \mathbb{Z}_p$, $Z \xleftarrow{R} \mathbb{G}_T$.

[2] The other is dual encryption system [Wat09].

Decisional Square Bilinear Diffie-Hellman Assumption. The decisional square bilinear Diffie- Hellman (DSBDH) assumption [Kil07] is that the distributions $(g, g^a, g^b, e(g, g)^{a^2 b})$ and (g, g^a, g^b, Z) are computationally indistinguishable, where $g \xleftarrow{R} \mathbb{G}^*$, $a, b \xleftarrow{R} \mathbb{Z}_p$, $Z \xleftarrow{R} \mathbb{G}_T$.

Decisional Truncated Augmented Bilinear Diffie-Hellman Exponent Assumption. The decisional truncated augmented bilinear Diffie-Hellman exponent (DTABDHE) assumption [Gen06] with respect to q is that the distributions $(g', g'_{q+2}, g, g_1, \dots, g_q, e(g_{q+1}, g'))$ and $(g', g'_{q+2}, g, g_1, \dots, g_q, Z)$ (here g_i and g'_i denote g^{α^i} and g'^{α^i}) are computationally indistinguishable, where $g, g' \xleftarrow{R} \mathbb{G}^*$, $\alpha \xleftarrow{R} \mathbb{Z}_p$, $Z \xleftarrow{R} \mathbb{G}_T$.

Quadratic Residuosity Assumption. Let $\mathsf{RSAGen}(\kappa)$ be a PPT algorithm that generates two κ-bit primes p and q. For a positive integer N, let $J(N)$ denote the set $J(N) = \{x \in \mathbb{Z}_N : \left(\frac{x}{N}\right) = 1\}$ where $\left(\frac{x}{N}\right)$ denotes the Jacobi symbol of x in $\mathbb{Z}_N$. Let $QR(N)$ denote the set of quadratic residues in $J(N)$. The quadratic residuosity (QR) assumption is that the distributions $(N \leftarrow pq, x \xleftarrow{R} QR(N))$ and $(N \leftarrow pq, x \xleftarrow{R} J(N) \backslash QR(N))$ are computationally indistinguishable.

Decisional Learning With Errors Assumption. For an integer $q \geq 2$ and some probability distribution χ over $\mathbb{Z}_q$, an integer dimension $n \in \mathbb{Z}^+$, and a vector $\mathbf{s} \in \mathbb{Z}_q^n$, define $A_{\mathbf{s},\chi}$ as the distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ of the variable $(\mathbf{a}, \mathbf{a}^T s + t)$ where $\mathbf{a} \xleftarrow{R} \mathbb{Z}_q^n$ and $t \leftarrow \chi$. The decisional learning with errors (DLWE) assumption is that the distribution $A_{\mathbf{s},\chi}$ and the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ are computationally indistinguishable, where $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q$.

2.3 Min-Entropy and Randomness Extractor

Here we review some concepts related to probability distributions and randomness extractors.

The *statistical distance* between two random variables x, y over a finite domain Ω is defined as $\mathbf{SD}(x, y) = \frac{1}{2} \sum_{\omega \in \Omega} |\Pr[x = \omega] - \Pr[y = \omega]|$. We say that two variables are *ϵ-close* if their statistical distance is at most ϵ.

The *min-entropy* of a random variable x over a domain Ω is the negative (base-2) logarithm of the *predictability* of x: $\mathbf{H}_\infty(x) = -\log_2(\max_{\omega \in \Omega} \Pr[x = \omega])$. In many scenarios, the variable x is correlated with another variable y (define on domain Θ) whose value is known to an adversary. In such scenarios, it is more convenient to use the notion of *average min-entropy* [DORS08], which captures the *average predictability* of x given knowledge of y. This is formally defined as:

$$\widetilde{\mathbf{H}}_\infty(x|y) = -\log_2 \left(E_{\theta \leftarrow \Theta} \left[\max_{\omega \in \Omega} \Pr[x = \omega | y = \theta] \right] \right)$$

The following lemma bound on average min-entropy:

Lemma 2.1 ([DORS08]) *If y takes at most 2^r possible values and z is any random variable, then*

$$\widetilde{\mathbf{H}}_\infty(x|(y, z)) \geq \mathbf{H}_\infty(x|z) - r$$

In out constructions, we will mainly use strong randomness extractor [Sha02] in the setting of average min-entropy. The precise definition is as follows:

Definition 2.2 *A polynomial-time function $\mathsf{Ext} : \Omega \to \{0, 1\}^v$ is an average case (m, ϵ)-strong extractor with seeds set $S = \{0, 1\}^\mu$, if for all pairs of random variables (x, y) such that x is distributed over Ω and $\widetilde{\mathbf{H}}_\infty(x|y) \geq m$, $s \leftarrow U_S$, $r \leftarrow U_m$, we have $\mathbf{SD}((\mathsf{Ext}(x; s), s, y)), (r, s, y)) \leq \epsilon$.*

2.4 Leakage Model

So far, there are mainly two leakage models for leakage-resilient cryptographic schemes. One is called the "bounded-leakage model" [AGV09, NS09, ADW09, ADN$^+$10, CDRW10], which does not restrict the type of leakage, but has to bound the overall amount of leakage. The other is called the "continuous-leakage model" [CDH$^+$00, MR04, DP08, FKPR10], which only bounds the amount of leakage *per period* (as opposed to overall) by continually refreshing the secret keys, but has to impose additional non-trivial restrictions on the types of leakage. In this paper, we consider security and anonymity for encryption schemes in the bounded-leakage model.

3 Generalized Hash Proof System

As we have discussed in the introduction, HPS serves as a good framework to unify many PKE schemes based on decisional assumptions. However, its original definition seems a bit strict to encompass latticed-based PKE schemes which essentially implement the hash proof technique. Aiming for utmost generality, in this work we generalize the original paradigm of HPS. The main difference between our generalization and that of Cramer and Shoup [CS02] is that we introduce distribution distinguish problem instead of subset membership problem. In more details, the distribution distinguish problem can be viewed as a relaxed version of the subset membership problem. The relaxations comes from three aspects: 1) remove the restriction that the supports of two distributions must be disjointed; 2) remove the restriction that the two distributions are both uniform; 3) extend L from a single language to a collection of languages indexed by the set of public keys. Looking ahead, the first and second relaxations admits more natural and efficient constructions for languages related to a broad class of assumptions, while the third relaxation admits more PKE constructions (e.g. the PKE schemes transformed from the IBE schemes presented in [CDRW10, Gen06]).

Distribution Distinguish Problem (DDP). A distribution distinguish problem $\mathbf{D}$ specifies a collection $(D_\kappa)_{\kappa\geq 0}$ of distributions. For each security parameter $\kappa \geq 0$, D_κ is a probability distribution over problem instance descriptions. Each instance description Γ specifies:

- Finite non-empty sets X, W, PK, and two collections of distributions $A = (A_{pk})_{pk\in PK}$ and $B = (B_{pk})_{pk\in PK}$ over X.
- A collection of binary relations $\mathsf{R} = (\mathsf{R}_{pk})_{pk\in PK}$ defined over $X \times W$. For $x \in X$ and $w \in W$ and some $pk \in PK$ such that $(x, w) \in \mathsf{R}_{pk}$, we say that w is a *witness* for x. We require that for all $x \in |A_{pk}|$, there exists a $w \in W$ such that $(x, w) \in \mathsf{R}_{pk}$.

We write $\Gamma = (X, W, PK, A, B, \mathsf{R})$ to indicate that the instance Γ specifies X, W, PK, A, B, and R as above. $\mathbf{D}$ also provides the three sampling algorithms below:

- $\mathsf{SampDDP}(\kappa)$: take as input a security parameter κ, output a master public and secret key pair (mpk, msk) and an instance description Γ according to the distribution D_κ. Here, mpk will be implicitly used by the following two sampling algorithm.
- $\mathsf{SampA}(pk)$: output $x \leftarrow A_{pk}$ along with a witness $w \in W$ such that $(x, w) \in \mathsf{R}_{pk}$. This is the *sampling with witness algorithm.*
- $\mathsf{SampB}(pk)$: output $x \leftarrow B_{pk}$. This is the *sampling without witness algorithm.*

We only require algorithms $\mathsf{SampDDP}$ and SampA to be efficient. A distribution distinguish problem $\mathbf{D}$ is said to be hard if A_{pk} and B_{pk} are computationally indistinguishable for any PPT adversary (a precise definition will be given later).

Projective Hash Family (PHF). Let X, Y, SK, PK be finite non-empty sets, and A be a collection of distributions indexed by PK. Here X, PK, A are defined as in DDP above. Let $\mathsf{H} = \{\mathsf{H}_{sk} : X \to Y\}_{sk \in SK}$ be a family of functions indexed by SK. Let $\alpha : SK \to PK$ be a projection from SK to PK. We say $\mathbf{H} = (\mathsf{H}, SK, PK, X, A, Y, \alpha)$ a projective hash family (PHF) if for any $sk \in SK$ and $pk = \alpha(sk)$, the action of H_{sk} on $x \leftarrow A_{pk}$ is approximately determined by $\alpha(sk)$ (in a sense we will make precise below).

Generalized Hash Proof System. A generalized hash proof system $\mathbf{P}$ bridges a distribution distinguish problem $\mathbf{D}$ and a projective hash family $\mathbf{H}$ with the following four algorithms:

- $\mathsf{Setup}(\kappa)$: run $\mathsf{SampDDP}(\kappa)$ to generate a master public/secret key pair (mpk, msk) and an instance description $\Gamma = (X, W, PK, A, B, \mathsf{R})$ of $\mathbf{D}$, pick a suitable projective hash family $\mathbf{H} = (\mathsf{H}, SK, PK, X, A, Y, \alpha)$. msk will be implicitly used by the following algorithms.
- $\mathsf{KeyGen}(\kappa)$: pick $sk \xleftarrow{R} SK$, compute $pk \leftarrow \alpha(sk)$, and output a public/secret key pair (pk, sk).
- $\mathsf{Priv}(sk, x)$: take as input a private key sk and $x \in X$, and output $y \in Y$ such that $y = \mathsf{H}_{sk}(x)$. This is the *private evaluation algorithm.*
- $\mathsf{Pub}(pk, x, w)$: take as input pk and $x \in |A_{pk}|$ together with a witness $w \in W$ for x, and output $y \in Y$. This is the *public evaluation algorithm.*

We now are ready to define soem properties for generalized HPS. Unless otherwise indicated, all the following properties are defined in the probability space of generating (mpk, msk) honestly by $\mathsf{SampDDP}(\kappa)$.

Definition 3.1 (Indistinguishability) *A HPS satisfies the indistinguishability if for any PPT adversary $\mathcal{A}$:*

$$|\Pr[\mathcal{A}(mpk, pk, sk, x) = 1 : x \leftarrow A_{pk}] - \Pr[\mathcal{A}(mpk, pk, sk, x) = 1 : x \leftarrow B_{pk}]| \leq \mathsf{negl}(\kappa)$$

where $(mpk, msk) \leftarrow \mathsf{SampDDP}(\kappa)$ and $(pk, sk) \leftarrow \mathsf{KeyGen}(\kappa)$.

This property captures the hardness of the underlying DDP.

Definition 3.2 (Projection) *A HPS is projective if:*

$$\Pr[\mathsf{Pub}(pk, x, w) \neq \mathsf{Priv}(sk, x)] \leq \mathsf{negl}(\kappa)$$

where w is a witness for x. The probability is taken over the choice of $x \leftarrow A_{pk}$ and $(pk, sk) \leftarrow \mathsf{KeyGen}(\kappa)$.

The above definition is in fact approximate projection, which is analogous to the notion of approximate correctness introduced in [KV09]. The property of approximate projection captures the average-case behavior of H_{sk} on $x \leftarrow A_{pk}$.

Definition 3.3 (Smoothness) *A HPS is ϵ-smooth if:*

$$\mathbf{SD}((R, pk, x, y), (R, pk, x, y')) \leq \epsilon$$

where R is the ensemble of (mpk, msk). The probability is taken over the choice $(pk, sk) \leftarrow \mathsf{KeyGen}(\kappa)$, $x \leftarrow B_{pk}$, $y \leftarrow \mathsf{Priv}(sk, x)$, $y' \leftarrow U_Y$. Moreover, a HPS is ℓ-leakage-resilient ϵ-smooth if for any function $f(\cdot)$[3] with ℓ-bit output:

$$\mathbf{SD}((R, pk, f(sk), x, y), (R, pk, f(sk), x, y')) \leq \epsilon$$

[3] Throughout this paper, leakage function $f(\cdot)$ is possibly randomized and need not be efficient, and we assume the output of $f(\cdot)$ can be equivalently obtained in an adaptive manner.

The property of smoothness captures the average-case behavior of H_{sk} on $x \leftarrow B_{pk}$.

Definition 3.4 (Anonymity) *A HPS is ϵ-anonymous if:*

$$\mathbf{SD}((R, pk_0, pk_1, x_0, y_0), (R, pk_0, pk_1, x_1, y_1)) \leq \epsilon$$

where R is the ensemble of (mpk, msk). The probability is taken over the choice of $(pk_i, sk_i) \leftarrow \mathsf{KeyGen}(\kappa)$, $x_i \leftarrow B_{pk_i}$, $y_i \leftarrow \mathsf{Priv}(sk_i, x_i)$ for $i = \{0, 1\}$. Moreover, a HPS is ℓ-leakage-resilient ϵ-anonymous if for any function $f_0(\cdot)$ and $f_1(\cdot)$ with ℓ-bit output:

$$\mathbf{SD}((R, pk_0, pk_1, f_0(sk_0), f_1(sk_1), x_0, y_0), (R, pk_0, pk_1, f_0(sk_0), f_1(sk_1), x_1, y_1)) \leq \epsilon$$

The property of anonymity captures the behavior of SampB from pairwise aspect.

3.1 Generic Construction of Leakage-Resilient HPS

We now show how to convert a HPS (Setup, KeyGen, Pub, Priv) into a leakage-resilient one using an average-case randomness extractor $\mathsf{Ext} : Y \to \{0,1\}^v$ with seeds set $S = \{0,1\}^\mu$. We slightly modify the algorithms SampA and SampB of $\mathbf{D}$ as follows:

- $\overline{\mathsf{SampA}}(pk)$: sample $(x, w) \leftarrow \mathsf{SampA}(pk)$, pick a seed $s \xleftarrow{R} \{0,1\}^\mu$, and output $\bar{x} = (x, s)$.
- $\overline{\mathsf{SampB}}(pk)$: sample $x \leftarrow \mathsf{SampB}(pk)$, pick a seed $s \xleftarrow{R} \{0,1\}^\mu$, and output $\bar{x} = (x, s)$.

We keep algorithms Setup and KeyGen unchanged, define:

- $\overline{\mathsf{Priv}}(sk, \bar{x})$: parse $\bar{x}$ as (x, s), compute $y \leftarrow \mathsf{Priv}(id, x)$, and output $\bar{y} \leftarrow \mathsf{Ext}(y; s)$.
- $\overline{\mathsf{Pub}}(pk, \bar{x}, w)$: parse $\bar{x}$ as (x, s), compute $y \leftarrow \mathsf{Pub}(pk, x, w)$, and output $\bar{y} \leftarrow \mathsf{Ext}(y; s)$.

The theorem below shows that the transformed HPS (Setup, KeyGen, $\overline{\mathsf{Pub}}$, $\overline{\mathsf{Priv}}$) is leakage-resilient for appropriate parameters.

Theorem 3.5 *Assume that the underlying HPS is ϵ_1-smooth and ϵ_2-anonymous and $|Y| = 2^m$. Let $\mathsf{Ext} : Y \to \{0,1\}^v$ be an average-case $(m - \ell, \epsilon_{\mathsf{ext}})$ randomness extractor. Then the above transform produces an ℓ-leakage-resilient $(\epsilon_1 + \epsilon_{\mathsf{ext}})$-smooth and $(\epsilon_2 + 2\epsilon_{\mathsf{ext}})$-anonymous HPS.*

Proof. The smoothness of the underlying HPS means $\mathbf{SD}((R, pk, x, y), (R, pk, x, y')) \leq \epsilon_1$, which implies $\widetilde{\mathbf{H}}_\infty(y|(R, pk, x)) \approx \log_2 |Y| = m$. In the presence of leakage, an adversary can obtain at most ℓ bits of leakage from the private key sk (modelled as a random variable $f(sk)$ with 2^ℓ values). By Lemma 2.1 we have $\widetilde{\mathbf{H}}_\infty(y|(R, pk, f(sk), x)) \approx \widetilde{\mathbf{H}}_\infty(y|(R, pk, x)) - \ell = m - \ell$, therefore according to the definition of a $(m - \ell, \epsilon_{\mathsf{ext}})$ randomness extractor, we have $\mathbf{SD}((R, pk, f(sk), x, \bar{y}), (R, pk, f(sk), x, \bar{y}')) \leq \epsilon_1 + \epsilon_{\mathsf{ext}}$. The leakage-resilient smoothness of the resulting HPS immediately follows the fact that s is independently chosen from $\{0,1\}^\mu$, that is:

$$\mathbf{SD}((R, pk, f(sk), \bar{x}, \bar{y}), (R, pk, f(sk), \bar{x}, \bar{y}')) \leq \epsilon_1 + \epsilon_{\mathsf{ext}}$$

This part of theorem has been implicitly implied in [NS09].

Next we prove the leakage-resilient anonymity of the resulting HPS. We first apply the leakage-resilient smooth property twice:

$$\mathbf{SD}((R, pk_0, f_0(sk_0), \bar{x}_0, \bar{y}_0), (R, pk_0, f_0(sk_0), \bar{x}_0, \bar{y}')) \leq \epsilon_{\mathsf{ext}}$$
$$\mathbf{SD}((R, pk_1, f_1(sk_1), \bar{x}_1, \bar{y}_1), (R, pk_1, f_1(sk_1), \bar{x}_1, \bar{y}')) \leq \epsilon_{\mathsf{ext}}$$

Note that (pk_0, sk_0) and (pk_1, sk_1) are independently generated by $\mathsf{KeyGen}(\kappa)$, then we have:

$$\mathbf{SD}((R, pk_0, pk_1, f_0(sk_0), f_1(sk_1), \bar{x}_0, \bar{y}_0), (R, pk_0, pk_1, f_0(sk_0), f_1(sk_1), \bar{x}_0, \bar{y}')) \leq \epsilon_{\mathsf{ext}} \tag{1}$$
$$\mathbf{SD}((R, pk_0, pk_1, f_0(sk_0), f_1(sk_1), \bar{x}_1, \bar{y}_1), (R, pk_0, pk_1, f_0(sk_0), f_1(sk_1), \bar{x}_1, \bar{y}')) \leq \epsilon_{\mathsf{ext}} \tag{2}$$

The anonymity of the underlying HPS means $\mathbf{SD}((R, pk_0, pk_1, x_0, y_0), (R, pk_0, pk_1, x_1, y_1)) \leq \epsilon_2$, thus certainly $\mathbf{SD}((R, pk_0, pk_1, x_0), (R, pk_0, pk_1, x_1)) \leq \epsilon_2$. The fact that $s \xleftarrow{R} \{0,1\}^\mu$ and $\bar{y}' \xleftarrow{R} \{0,1\}^v$ are independent of x_0, x_1 further implies $\mathbf{SD}((R, pk_0, pk_1, \bar{x}_0, \bar{y}'), (R, pk_0, pk_1, \bar{x}_1, \bar{y}')) \leq \epsilon_2$. Note that conditioned on fixed pk_0 and pk_1, $f_0(sk_0)$ and $f_1(sk_1)$ are independent of $\bar{x}_0$, $\bar{x}_1$, we then arrive at:

$$\mathbf{SD}((R, pk_0, pk_1, f_0(sk_0), f_1(sk_1), \bar{x}_0, \bar{y}'), (R, pk_0, pk_1, f_0(sk_0), f_1(sk_1), \bar{x}_1, \bar{y}')) \leq \epsilon_2 \tag{3}$$

The leakage-resilient anonymity of the resulting HPS immediately follows by combining the inequalities 1, 2, 3, that is:

$$\mathbf{SD}((R, pk_0, pk_1, f_0(sk_0), f_1(sk_1), \bar{x}_0, \bar{y}_0), (R, pk_0, pk_1, f_0(sk_0), f_1(sk_1), \bar{x}_1, \bar{y}_1)) \leq \epsilon_2 + 2\epsilon_{\mathsf{ext}}$$

This proves the theorem. □

4 Leakage-Resilient PKE

A PKE scheme consists of three PPT algorithms as below:

- $\mathsf{KeyGen}(\kappa)$: take as input a security parameter κ, output a public/secret key pair (pk, sk). Let M be the message space, and C be the ciphertext space.
- $\mathsf{Encrypt}(pk, m)$: take as input a public key pk and a message $m \in M$, output a ciphertext c.
- $\mathsf{Decrypt}(sk, c)$: take as input a secret key sk and a ciphertext $c \in C$, output a message m or a reject symbol $\perp$ indicating c is invalid.

The correctness of PKE requires that for any message $m \in M$, we have:

$$\Pr[\mathsf{Decrypt}(sk, \mathsf{Encrypt}(pk, m)) \neq m] \leq \mathsf{negl}(\kappa)$$

where the probability is taken over the random coins used by KeyGen and $\mathsf{Encrypt}$.

Security. The data privacy we consider for PKE is leakage-resilient IND-CPA security in the bounded-leakage model. Advantage of an adversary $\mathcal{A}$ is defined as:

$$\mathrm{Adv}_{\mathcal{A}}(\kappa) = \Pr\left[b = b' : \begin{array}{l} (pk, sk) \leftarrow \mathsf{KeyGen}(\kappa); \\ m \leftarrow \mathcal{A}^{\mathcal{O}_{\text{leak}}(sk, \cdot)}(pk); \\ b \xleftarrow{R} \{0,1\}; \\ c^* \leftarrow \mathsf{Encrypt}(pk, m); \\ b' \leftarrow \mathcal{A}(c^*) \end{array} \right] - \frac{1}{2}$$

where $\mathcal{O}_{\text{leak}}(sk, \cdot)$ is an oracle that on input a function $f : SK \to \{0,1\}^*$, returns $f(sk)$. We say $\mathcal{A}$ an ℓ-leakage adversary if the sum of output length of all functions that it submits to the leakage oracle $\mathcal{O}_{\text{leak}}(sk, \cdot)$ is less than ℓ. A PKE scheme is said to be ℓ-leakage-resilient IND-CPA secure if for any PPT ℓ-leakage adversary $\mathcal{A}$, its advantage defined as above is negligible in κ.

Anonymity. The key privacy we consider for PKE is leakage-resilient ANO-CPA anonymity in the bounded-retrieval model. Advantage of an adversary $\mathcal{A}$ is defined as:

$$\mathrm{Adv}_{\mathcal{A}}(\kappa) = \Pr\left[b = b' : \begin{array}{l} (pk_0, sk_0) \leftarrow \mathsf{KeyGen}(\kappa), (pk_1, sk_1) \leftarrow \mathsf{KeyGen}(\kappa); \\ m \leftarrow \mathcal{A}^{\mathcal{O}_{\text{leak}}(sk_0,\cdot),\mathcal{O}_{\text{leak}}(sk_1,\cdot)}(pk_0, pk_1); \\ b \xleftarrow{R} \{0,1\}; \\ c^* \leftarrow \mathsf{Encrypt}(pk_b, m); \\ b' \leftarrow \mathcal{A}(c^*) \end{array} \right] - \frac{1}{2}$$

where $\mathcal{O}_{\text{leak}}(sk, \cdot)$ is an oracle that on input a function $f : SK \to \{0,1\}^*$, returns $f(sk)$. We say $\mathcal{A}$ an ℓ-leakage adversary if the sum of output length of all functions that it submits to the leakage oracle $\mathcal{O}_{\text{leak}}(sk_0, \cdot)$ and $\mathcal{O}_{\text{leak}}(sk_1, \cdot)$ are both less than ℓ. A PKE scheme is said to be ℓ-leakage-resilient ANO-CPA anonymous if for any PPT ℓ-leakage adversary $\mathcal{A}$, its advantage defined as above is negligible in κ.

The construction of a leakage-resilient PKE from a leakage-resilient HPS is straightforward. Starting from a leakage-resilient HPS where the hash value set Y has some group structure $(Y, +)$ (e.g. bit-strings with "exclusive or" $\oplus$), we can construct a PKE scheme with the common public parameters mpk and the message set $M = Y$ by simply using the hashing value as one-time-pad to mask message. More precisely:

- $\mathsf{KeyGen}(\kappa)$: the same as that in HPS.
- $\mathsf{Encrypt}(pk, m)$: $(x, w) \leftarrow \mathsf{SampA}(pk)$, compute $y \leftarrow \mathsf{Pub}(pk, x, w)$, set $z = y + m$, and output $c = (x, z)$.
- $\mathsf{Decrypt}(sk, c)$: parse c as (x, z), compute $y \leftarrow \mathsf{Priv}(sk, c)$, and output $m = z - y$.

The correctness of the resulting PKE scheme follows from the projective property of the starting HPS. Note that the algorithm SampB is not used in the construction of PKE, but it will be used to establish security.

Theorem 4.1 *The above construction yields an ℓ-leakage-resilient IND-CPA secure PKE if the underlying HPS is ℓ-leakage-resilient smooth.*

Proof. The proof of this theorem is rather straightforward and follows from the same argument presented in [CS02, NS09]. □

Theorem 4.2 *The above construction yields an ℓ-leakage-resilient ANO-CPA anonymous PKE if the underlying HPS is ℓ-leakage-resilient anonymous.*

Proof. We proceed via a sequence of games.

Game 0: Define Game 0 as the leakage-resilient anonymous game for PKE. In the challenge stage of Game 0, upon receiving a message m submitted by the adversary, the challenger picks a random bit b and computes $c_b \leftarrow \mathsf{Encrypt}(pk_b, m)$. We expands c_b as (x_b, z_b) where:

$$(x_b, w) \leftarrow \mathsf{SampA}(pk_b), y_b \leftarrow \mathsf{Pub}(pk_b, x_b, w), z_b = y_b + m$$

Game 1: Compared to Game 0, we modify the challenge stage by letting the challenger generate the ciphertext $c_b = (x_b, z_b)$ using the private key sk_b of pk_b:

$$(x_b, w) \leftarrow \mathsf{SampA}(pk_b), \tilde{y}_b \leftarrow \mathsf{Priv}(sk_b, x_b), z_b = \tilde{y}_b + m$$

The difference between Game 0 an Game 1 is only using $\tilde{y}_b$ in the place of y_b. By the approximate projection of the underlying HPS, the probability $\tilde{y}_b \neq y_b$ is negligible in κ. Thereby, Game 0 and Game 1 are indistinguishable.

Game 2: Based on Game 1, we further modify the challenge stage by letting the challenger generate the ciphertext $c_b = (x_b, z_b)$ as follows:

$$x_b \leftarrow \mathsf{SampB}(pk_b), \tilde{y}_b \leftarrow \mathsf{Priv}(sk_b, x_b), z_b = \tilde{y}_b + m$$

We argue that Game 1 and Game 2 are computationally indistinguishable by the indistinguishability of the underlying IB-HPS. Although the definition of indistinguishability does not explicitly embody private key leakage queries, it allows the adversary to learn the entire private keys. Therefore, Game 1 and Game 2 are indistinguishable even if the adversary obtains the entire information of private keys for the two target public keys, and thus certainly being indistinguishable when the adversary is only given limited amount of leakage.

According to the ℓ-leakage-resilient anonymous property of HPS, the advantage of any PPT adversary in Game 2 is negligible. Therefore the advantage of any PPT adversary in Game 0 is also negligible in κ, which concludes the Theorem 4.2. □

5 Instantiations of HPS

Previous constructions of HPS based on a variety of assumptions [CS02, NS09] can also be explained in the paradigm of generalized HPS without any difficulty. To avoid repetition, we only present a construction of HPS based on the DLWE assumption, which does not fit the original paradigm.

5.1 HPS based on the DLWE Assumption

We now describe a HPS based on the DLWE assumption, which can be viewed as the backbone of the PKE scheme presented in [GPV08].

Let $\mathbf{D}$ be a distribution distinguish problem based on the DLWE assumption. $\mathsf{SampDDP}(\kappa)$ generates $\mathbb{A} \in \mathbb{Z}_q^{n \times m}$ along with a trapdoor $\mathbb{S} \subset \Lambda^{\perp}(\mathbb{A}, q)$ according to the trapdoor generation algorithm of [GPV08], and a function f indexed by $\mathbb{A}$, sets $mpk = (\mathbb{A}, f_{\mathbb{A}})$ and $msk = \mathbb{S}$; outputs an instance description $\Gamma = (X, W, PK, A, B, \mathsf{R})$ of $\mathbf{D}$, where $X = \mathbb{Z}_q^n \times \mathbb{Z}_q$, $W = \mathbb{Z}_q^n$, $PK = \mathbb{Z}_q^n$, $\mathsf{R} = \{((p, v), w) \in X \times W : ((\mathbb{A}^T w + t, v), w)\}$ where $t \in \chi^m$, $v \in \mathbb{Z}_p\}$, two collections of distributions A and B are specified by SampA and SampB as follows:

- $\mathsf{SampA}(pk)$: pick $w \xleftarrow{R} \mathbb{Z}_q^n$, $t \xleftarrow{R} \chi^m$, $v \xleftarrow{R} \mathbb{Z}_q$, compute $p = \mathbb{A}^T w + t$, output $x = (p, v)$ and $w \in W$.
- $\mathsf{SampB}(pk)$: pick $p \xleftarrow{R} \mathbb{Z}_q^m$ and $v \xleftarrow{R} \mathbb{Z}_q$, output $x = (p, v)$.

In this case, A_{pk}, B_{pk}, and R_{pk} are the same for all $pk \in PK$. We then write A, B, and R for simplicity.

Let $\mathbf{H} = (\mathsf{H}, SK, PK, X, Y, A, \alpha)$ be a corresponding projective hash family, where $SK = \mathbb{Z}_q^m$, $Y = \mathbb{Z}_2 \times \mathbb{Z}_q$, X, PK, and A are defined as above. For $x = (p, v)$, we define $\mathsf{H}_{sk}(x) = y$ as $y = 1$ if $|v - sk^T p| \leq \frac{q-1}{4}$ and $y = 0$ otherwise.

Let $\mathbf{P}$ be an HPS for $\mathbf{D}$ associating $\mathbf{H}$, which consists of four algorithms as below:

- $\mathsf{Setup}(\kappa)$: run $\mathsf{SampDDP}(\kappa)$ to generate $mpk = (\mathbb{A}, f_{\mathbb{A}})$ and $msk = \mathbb{S}$.
- $\mathsf{KeyGen}(\kappa)$: choose an error vector according to $D_{\mathbb{Z}_q^m, r}$ as the secret key sk, set the public key pk to be $f_{\mathbb{A}}(sk)$.
- $\mathsf{Priv}(sk, x)$: parse $x = (p, v)$, if $|v - sk^T p| \leq \frac{q-1}{4}$ then output $y = 1$ else output $y = 0$.
- $\mathsf{Pub}(pk, x, w)$: parse $x = (p, v)$, if $|v - pk^T w| \leq \frac{q-1}{4}$ then output $y = 1$ else output $y = 0$.

As shown in [GPV08], the above HPS is smooth and anonymous.

6 Generalized Identity-Based Hash Proof System

The paradigm of IB-HPS has appeared in different forms in previous literature [BGH07, ADN$^+$10]. In [BGH07], IB-HPS is viewed as HPS with trapdoor. However, their definition inherently relates to the original SMP, hence is not generic enough to encompass all the IBE schemes relying on hash proof techniques, e.g. the IBE schemes presented in [Gen06, CDRW10]. In [ADN$^+$10], IB-HPS is viewed as IB-KEM with some special algebraic properties. Although their definition extended the original SMP by making the encapsulation algorithms take a public key as input, they still retained the "disjointed" requirement by equipping with valid and invalid encapsulation. In [ADN$^+$10], they also introduced a property called "anonymous-encapsulation". Briefly, anonymous-encapsulation requires that there exists an efficient and equivalent encapsulation algorithm that can generate the ciphertext and the DEM key without knowing the intended identity. As they summarized, the anonymity-encapsulation property brings two advantages. Firstly, IB-HPS with such property immediately implies anonymous IBE, in that the ciphertext does not contain any information of the intended identity. Secondly, IB-HPS with such property allows the resulting leakage-amplification scheme to greatly reduce the ciphertext size by adapting the standard randomness reuse trick (here witness serves as randomness). We remark that anonymous-encapsulation of IB-HPS is a sufficient but not a necessary condition for anonymity of the resulting IBE. For instance, as we will see in subsection 8.4, Gentry IBE [Gen06] is anonymous but its underlying IB-HPS does not satisfy anonymous-encapsulation property.

In what follows, parallel to what we did in section 3, we generalize the paradigm of IB-HPS. and then introduce a property named anonymity to serve as a sufficient and necessary condition for anonymity of the resulting IBE. As we will see later, our paradigm of anonymous (IB)-HPS serves as a good framework to explain all the known anonymous encryption schemes [Gen06, BGH07, GPV08, Cor09] that rely on hash proof technique, either in the random oracle model or in the standard model.

6.1 Definition of Generalized IB-HPS

The definition of distribution distinguish problem and projective hash family are the same as that of HPS. An identity-based hash proof system (IB-HPS) $\mathbf{P}$ bridges DDP $\mathbf{D}$ and PHF $\mathbf{H}$ with the following four algorithms:

- $\mathsf{Setup}(\kappa)$: the same as that in HPS; in addition, it aslo specifies an identity set I, a function $\mathsf{IHF} : I \to PK$, and requires that α can be efficiently inverted with msk. We denote its inverse function by $\sigma(msk, \cdot)$, which satisfies $\alpha(\sigma(msk, pk)) = pk$ for any $pk \in PK$.
- $\mathsf{KeyGen}(msk, id)$: take as input msk and $id \in I$, and output $sk \leftarrow \sigma(msk, \mathsf{IHF}(id))$.
- $\mathsf{Priv}(sk, x)$: the same as that in HPS. This is the *private evaluation algorithm*.
- $\mathsf{Pub}(id, x, w)$: take as input $id \in I$, $x \in L_{pk}$ (where $pk = \mathsf{IHF}(id)$) and a witness $w \in W$ for x, and output $y \in Y$. This is the *public evaluation algorithm*.

We then define the indistinguishability, (approximate) projection, smoothness, and anonymity for IB-HPS. Unless otherwise indicated, all the following properties are defined in the probability space of generating (mpk, msk) honestly by $\mathsf{SampDDP}(\kappa)$.

Definition 6.1 (Indistinguishability) *An IB-HPS satisfies the indistinguishability if for any PPT adversary $\mathcal{A}$ and any $id \in I$,*

$$|\Pr[\mathcal{A}^{\mathcal{O}_{\text{reveal}}(\cdot)}(mpk, id, x) = 1 : x \leftarrow A_{pk}] - \Pr[\mathcal{A}^{\mathcal{O}_{\text{reveal}}}(mpk, id, x) = 1 : x \leftarrow B_{pk}]| \leq \mathsf{negl}(\kappa)$$

where $pk = \mathsf{IHF}(id)$, $\mathcal{O}_{\text{reveal}}(\cdot)$ is an oracle that on input of $id \in I$, returns $sk \leftarrow \mathsf{KeyGen}(msk, id)$. The probability is taken over the choice of $(mpk, msk) \leftarrow \mathsf{SampDDP}(\kappa)$.

Definition 6.2 (Projection) *An IB-HPS is projective if for any $id \in I$:*

$$\Pr[\mathsf{Pub}(id, x, w) \neq \mathsf{Priv}(sk, x)] \leq \mathsf{negl}(\kappa)$$

where w is a witness for x. The probability is taken over the choice of $x \leftarrow A_{pk}$ ($pk = \mathsf{IHF}(id)$) and $sk \leftarrow \mathsf{KeyGen}(msk, id)$.

Definition 6.3 (Smoothness) *An IB-HPS is ϵ-smooth if for any $id \in I$:*

$$\mathbf{SD}((R, id, x, y), (R, id, x, y')) \leq \epsilon$$

where R is the ensemble of (mpk, msk) and the private keys for identities other than id. The probability is taken over the choice of $x \leftarrow B_{pk}$ for $pk \leftarrow \mathsf{IHF}(id)$, $y \leftarrow \mathsf{Priv}(sk, x)$ ($sk \leftarrow \mathsf{KeyGen}(msk, id)$), and $y' \leftarrow U_Y$. Moreover, an IB-HPS is ℓ-leakage-resilient ϵ-smooth if for any function $f(\cdot)$ with ℓ-bit output:

$$\mathbf{SD}((R, id, f(sk), x, y), (R, id, f(sk), x, y')) \leq \epsilon$$

Definition 6.4 (Anonymity) *An IB-HPS is ϵ-anonymous if for any two distinct $id_0, id_1 \in I$:*

$$\mathbf{SD}((R, id_0, id_1, x_0, y_0), (R, id_0, id_1, x_1, y_1)) \leq \epsilon$$

where R is the ensemble of (mpk, msk) and the private keys for identities other than id_0 and id_1. The probability is taken over the choice of $x_i \leftarrow B_{pk_i}$ for $pk_i \leftarrow \mathsf{IHF}(id_i)$), $y_i \leftarrow \mathsf{Priv}(sk_i, x_i)$ ($sk_i \leftarrow \mathsf{KeyGen}(msk, id_i)$) for $i \in \{0, 1\}$, Moreover, an IB-HPS is ℓ-leakage-resilient ϵ-anonymous if, for any function $f_0(\cdot)$ and $f_1(\cdot)$ with ℓ-bit output, we have:

$$\mathbf{SD}((R, id_0, id_1, f_0(sk_0), f_1(sk_1), x_0, y_0), (R, id_0, id_1, f_0(sk_0), f_1(sk_1), x_1, y_1)) \leq \epsilon$$

The generic construction of leakage-resilient IB-HPS from IB-HPS and the generic construction of leakage-resilient IBE from leakage-resilient IB-HPS are similar to that in PKE setting. For completeness, we present the technical details as below.

6.2 Generic Construction of Leakage-Resilient IB-HPS

We now show a generic construction of leakage-resilient IB-HPS, which is almost the same as that presented in [ADN+10, Section 3.2]. Starting from an IB-HPS (Setup, KeyGen, Pub, Priv), the construction converts it to a leakage-resilient IB-HPS using an average-case randomness extractor $\mathsf{Ext} : Y \to \{0, 1\}^v$ with seeds set $\{0, 1\}^\mu$. We slightly modify the algorithms SampA and SampB of $\mathbf{D}$ as follows:

- $\overline{\mathsf{SampA}}(pk)$: sample $(x, w) \leftarrow \mathsf{SampA}(pk)$, pick a seed $s \xleftarrow{R} \{0, 1\}^\mu$, and output $\bar{x} = (x, s)$.
- $\overline{\mathsf{SampB}}(pk)$: sample $x \leftarrow \mathsf{SampB}(pk)$, pick a seed $s \xleftarrow{R} \{0, 1\}^\mu$, and output $\bar{x} = (x, s)$.

We keep algorithms Setup and KeyGen unchanged, define:

- $\overline{\mathsf{Priv}}(sk, \bar{x})$: parse $\bar{x}$ as (x, s), compute $y \leftarrow \mathsf{Priv}(id, x)$, and output $\bar{y} = \mathsf{Ext}(y; s)$.
- $\overline{\mathsf{Pub}}(id, \bar{x}, w)$: parse $\bar{x}$ as (x, s), compute $y \leftarrow \mathsf{Pub}(id, x, w)$, and output $\bar{y} = \mathsf{Ext}(y; s)$.

The theorem below shows that the transformed IB-HPS (Setup, KeyGen, $\overline{\mathsf{Pub}}$, $\overline{\mathsf{Priv}}$) is leakage-resilient smooth and anonymous for appropriate parameters.

Theorem 6.5 *Assume that an IB-HPS is ϵ_1-smooth and ϵ_2-anonymous and $|Y| = 2^m$. Let $\mathsf{Ext} : Y \to \{0, 1\}^v$ be an $(m - \ell, \epsilon_{\mathsf{ext}})$ average-case randomness extractor. Then the above transform produces an ℓ-leakage-resilient $(\epsilon_1 + \epsilon_{\mathsf{ext}})$-smooth and $(\epsilon_2 + 2\epsilon_{\mathsf{ext}})$-anonymous IB-HPS.*

Proof. The smoothness of the underlying IB-HPS means that $\mathbf{SD}((R, id, x, y), (R, id, x, y')) \leq \epsilon_1$, which implies $\widetilde{\mathbf{H}}_\infty(y|(R, id, x)) \approx \log_2|Y| = m$. In the presence of leakage, an adversary has access to at most ℓ bits of leakage from the private key sk (modelled as a random variable $f(sk)$ with 2^ℓ values). By Lemma 2.1 we know that $\widetilde{\mathbf{H}}_\infty(y|(R, f(sk), id, x)) \approx \widetilde{\mathbf{H}}_\infty(y|(R, f(sk), id, x)) - \ell = m - \ell$, therefore according to the definition of a $(m - \ell, \epsilon_{\mathsf{ext}})$ randomness extractor, we have $\mathbf{SD}((R, id, f(sk), x, \bar{y}), (R, id, f(sk), x, \bar{y}')) \leq \epsilon_1 + \epsilon_{\mathsf{ext}}$. The leakage-resilient smoothness of the resulting IB-HPS follows from the fact that s is chosen independently from $\{0,1\}^\mu$, that is:

$$\mathbf{SD}((R, id, f(sk), \bar{x}, \bar{y}), (R, id, f(sk), \bar{x}, \bar{y}')) \leq \epsilon_1 + \epsilon_{\mathsf{ext}}$$

This part of theorem has been proved in another way in [ADN+10].

Next we prove leakage-resilient anonymity of the resulting IB-HPS. For any two distinct identities $id_0, id_1 \in I$, denote by R_0 (resp. R_1) the ensemble of mpk, msk, and the private keys for identities other than sk_0 (resp. sk_1). We first apply the leakage-resilient smooth property twice:

$$\mathbf{SD}((R_0, id_0, f_0(sk_0), \bar{x}_0, \bar{y}_0), (R_0, id_0, f_0(sk_0), \bar{x}_0, \bar{y}')) \leq \epsilon_{\mathsf{ext}}$$
$$\mathbf{SD}((R_1, id_1, f_1(sk_1), \bar{x}_1, \bar{y}_1), (R_1, id_1, f_1(sk_1), \bar{x}_1, \bar{y}')) \leq \epsilon_{\mathsf{ext}}$$

Note that $sk_1 \in R_0$ and $sk_0 \in R_1$, we have:

$$\mathbf{SD}((R_0, id_0, id_1, f_0(sk_0), f_1(sk_1), \bar{x}_0, \bar{y}_0), (R_0, id_0, id_1, f_0(sk_0), f_1(sk_1), \bar{x}_0, \bar{y}')) \leq \epsilon_{\mathsf{ext}}$$
$$\mathbf{SD}((R_1, id_0, id_1, f_0(sk_0), f_1(sk_1), \bar{x}_1, \bar{y}_1), (R_1, id_0, id_1, f_0(sk_0), f_1(sk_1), \bar{x}_1, \bar{y}')) \leq \epsilon_{\mathsf{ext}}$$

Note the fact that $R \subset R_0$ and $R \subset R_1$, then we get:

$$\mathbf{SD}((R, id_0, id_1, f_0(sk_0), f_1(sk_1), \bar{x}_0, \bar{y}_0), (R, id_0, id_1, f_0(sk_0), f_1(sk_1), \bar{x}_0, \bar{y}')) \leq \epsilon_{\mathsf{ext}} \quad (4)$$
$$\mathbf{SD}((R, id_0, id_1, f_0(sk_0), f_1(sk_1), \bar{x}_1, \bar{y}_1), (R, id_0, id_1, f_0(sk_0), f_1(sk_1), \bar{x}_1, \bar{y}')) \leq \epsilon_{\mathsf{ext}} \quad (5)$$

The anonymity of the underlying IB-HPS indicates $\mathbf{SD}((R, id_0, id_1, x_0, y_0), (R, id_0, id_1, x_1, y_1)) \leq \epsilon_2$ and thus certainly $\mathbf{SD}((R, id_0, id_1, x_0), (R, id_0, id_1, x_1)) \leq \epsilon_2$. The fact that $s \xleftarrow{R} \{0,1\}^\mu$ and $\bar{y}' \xleftarrow{R} \{0,1\}^v$ are independent of x_0, x_1 implies $\mathbf{SD}((R, id_0, id_1, \bar{x}_0, \bar{y}'), (R, id_0, id_1, \bar{x}_1, \bar{y}')) \leq \epsilon_1$. Note that conditioned on fixed id_0 and id_1, $f_0(sk_0)$ and $f_1(sk_1)$ are independent of $\bar{x}_0$, $\bar{x}_1$, we arrive at:

$$\mathbf{SD}((R, id_0, id_1, f_0(sk_0), f_1(sk_1), \bar{x}_0, \bar{y}'), (R, id_0, id_1, f_0(sk_0), f_1(sk_1), \bar{x}_1, \bar{y}')) \leq \epsilon_2 \quad (6)$$

Finally, the desired leakage-resilient anonymity of the IB-HPS immediately follows by combining the inequalities 4, 5, 6, that is:

$$\mathbf{SD}((R, id_0, id_1, f_0(sk_0), f_1(sk_1), \bar{x}_0, \bar{y}_0), (R, id_0, id_1, f_0(sk_0), f_1(sk_1), \bar{x}_1, \bar{y}_1)) \leq \epsilon_1 + 2\epsilon_{\mathsf{ext}}$$

This proves the theorem. □

7 Leakage-Resilient IBE

An IBE scheme [BF03] consists of four PPT algorithms as below:

- $\mathsf{Setup}(\kappa)$: take as input a security parameter κ, output a master public/secret key pair (mpk, msk). Let I be the identity space, M be the message space, and C be the ciphertext space. mpk will be used as an implicit input for algorithms $\mathsf{Extract}$, $\mathsf{Encrypt}$, and $\mathsf{Decrypt}$,

- $\mathsf{KeyGen}(msk, id)$: take as input msk and an identity $id \in I$, output a private key sk.
- $\mathsf{Encrypt}(id, m)$: take as input mpk, an identity $id \in I$, and a message $m \in M$, output a ciphertext $c \in C$.
- $\mathsf{Decrypt}(sk, c)$: take as input a private key sk and a ciphertext $c \in C$, output a message $m \in M$ or a reject symbol $\perp$ indicating c is invalid.

The correctness of IBE requires that for any $id \in I$ and any $m \in M$, we have:

$$\Pr[\mathsf{Decrypt}(\mathsf{KeyGen}(msk, id), \mathsf{Encrypt}(id, m)) \neq m] \leq \mathsf{negl}(\kappa)$$

where the probability is taken over random coins used by Setup, KeyGen, and $\mathsf{Encrypt}$.

Security. The data privacy we consider for IBE is leakage-resilient IND-CPA security in the bounded-leakage model. Advantage of an adversary $\mathcal{A}$ is defined as:

$$\mathrm{Adv}_{\mathcal{A}}(\kappa) = \Pr\left[b = b' : \begin{array}{l} (mpk, msk) \leftarrow \mathsf{Setup}(\kappa); \\ (m_0, m_1, id^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{reveal}}(\cdot), \mathcal{O}_{\text{leak}}(\cdot,\cdot)}(mpk); \\ b \xleftarrow{R} \{0,1\}; \\ c^* \leftarrow \mathsf{Encrypt}(id^*, m_b); \\ b' \leftarrow \mathcal{A}^{\mathcal{O}_{\text{reveal}}(\cdot), \mathcal{O}_{\text{leak}}(\cdot,\cdot)}(c^*) \end{array} \right] - \frac{1}{2}$$

where $\mathcal{O}_{\text{reveal}}(\cdot)$ is an oracle that on input id, returns $sk \leftarrow \mathsf{KeyGen}(msk, id)$, $\mathcal{O}_{\text{leak}}(\cdot, \cdot)$ is an oracle that on input id and a function $f : SK \to \{0,1\}^*$, returns $f(sk)$. The restrictions are that the reveal oracle is only available for identities other than id^*, and after seeing the challenge ciphertext c^* the leak oracle is only available for identities other than id^*. We say $\mathcal{A}$ an ℓ-leakage adversary if the sum of output length of all functions that it submits to the leakage oracle $\mathcal{O}_{\text{leak}}(id, \cdot)$ for any single $id \in I$ is less than ℓ. An IBE scheme is said to be ℓ-leakage-resilient IND-CPA secure if for any PPT ℓ-leakage adversary $\mathcal{A}$, its advantage defined as above is negligible in κ.

Anonymity. The key privacy we consider for IBE is leakage-resilient ANO-CPA security in the bounded-leakage model. Advantage of an adversary $\mathcal{A}$ is defined as:

$$\mathrm{Adv}_{\mathcal{A}}(\kappa) = \Pr\left[b = b' : \begin{array}{l} (mpk, msk) \leftarrow \mathsf{Setup}(\kappa); \\ (m, id_0, id_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{reveal}}(\cdot), \mathcal{O}_{\text{leak}}(\cdot,\cdot)}(mpk); \\ b \xleftarrow{R} \{0,1\}; \\ c^* \leftarrow \mathsf{Encrypt}(id_b, m); \\ b' \leftarrow \mathcal{A}^{\mathcal{O}_{\text{reveal}}(\cdot), \mathcal{O}_{\text{leak}}(\cdot,\cdot)}(c^*) \end{array} \right] - \frac{1}{2}$$

where $\mathcal{O}_{\text{reveal}}(\cdot)$ and $\mathcal{O}_{\text{leak}}(\cdot, \cdot)$ oracles are defined as above. The restrictions are that the reveal oracle is only available for identities other than id_0, id_1, and after seeing the challenge ciphertext c^* the leak oracle is only available for identities other than id_0, id_1. We say $\mathcal{A}$ an ℓ-leakage adversary if the sum of output length of all functions that it submits to the leakage oracle $\mathcal{O}_{\text{leak}}(id, \cdot)$ for any single $id \in I$ is less than ℓ. An IBE scheme is said to be ℓ-leakage-resilient ANO-CPA anonymous if for any PPT ℓ-leakage adversary $\mathcal{A}$, its advantage defined as above is negligible in κ.

Remark 1. Our notion of leakage-resilience only allows leakage on single private key for each identity, but not on master secret key.

We next show a natural construction of leakage-resilient IBE from IB-HPS, which is almost the same to the construction presented in [ADN+10, Section 4.2]. More precisely, given an

IB-HPS where the hashing value set Y has some group structure $(Y, +)$ (e.g. bit-strings with "exclusive or" $\oplus$), one can construct an IBE with identity set I and message set $M = Y$ by simply using the hashing value as one-time-pad to mask a message.. The algorithms Setup and KeyGen are identical to that of IB-HPS. The algorithms $\mathsf{Encrypt}$ and $\mathsf{Decrypt}$ are constructed as follows:

- $\mathsf{Encrypt}(id, m)$: compute $pk \leftarrow \mathsf{IHF}(id)$, $(x, w) \leftarrow \mathsf{SampA}(pk)$, $y \leftarrow \mathsf{Pub}(id, x, w)$, set $z = y + m$, and output $c = (x, z)$.
- $\mathsf{Decrypt}(sk, c)$: parse c as (x, z), compute $y \leftarrow \mathsf{Priv}(sk, x)$, and output $m = z - y$.

The correctness of the resulting PKE scheme follows from the projective property of the starting HPS. Note that the algorithm SampB of the IB-HPS is not used in the construction, but will be used to argue security.

Theorem 7.1 *The above construction yields an ℓ-leakage-resilient CPA-secure IBE if the underlying IB-HPS is ℓ-leakage-resilient smooth.*

Proof. The proof of this theorem has been presented in [ADN+10]. □

Theorem 7.2 *The above construction yields an ℓ-leakage-resilient anonymous IBE if the underlying IB-HPS is ℓ-leakage-resilient anonymous.*

Proof. We proceed via a sequence of games.

Game 0: Define Game 0 as the standard anonymous game for IBE. In the challenge stage of Game 0, upon receiving two identities id_0 and id_1 and a message m submitted by the adversary, the challenger picks a random bit b and computes $c_b \leftarrow \mathsf{Encrypt}(id_b, m)$. We expands c_b as (x_b, z_b) where:

$$(x_b, w) \leftarrow \mathsf{SampA}(id_b), y_b \leftarrow \mathsf{Pub}(id_b, x_b, w), z_b = y_b + m$$

Game 1: Compared to Game 0, we modify the challenge stage by having the challenger generate the ciphertext $c_b = (x_b, z_b)$ using the private key sk_b of id_b:

$$(x_b, w) \leftarrow \mathsf{SampA}(id_b), \tilde{y}_b \leftarrow \mathsf{Priv}(sk_b, x_b), z_b = \tilde{y}_b + m$$

The difference between Game 0 an Game 1 is only the use of $\tilde{y}_b$ versus y_b. By the approximate correctness of evaluation, $\tilde{y}_b \neq y_b$ happens with negligible probability, so Game 0 and Game 1 are indistinguishable.

Game 2: Based on Game 1, we further modify the challenge stage by having the challenger generate the ciphertext $c_b = (x_b, z_b)$ as follows:

$$x_b \leftarrow \mathsf{SampB}(id_b), \tilde{y}_b \leftarrow \mathsf{Priv}(sk_b, x_b), z_b = \tilde{y}_b + m$$

We claim that Game 1 and Game 2 are computationally indistinguishable by the distribution indistinguishability of the underlying IB-HPS. Note that, although the definition of distribution indistinguishability does not explicitly embody private key leakage queries, it allows the adversary to learn all private keys. Therefore indistinguishability between Game 1 and Game 2 holds even if the adversary obtains the entire information of private keys for the two target identities, and hence certainly holds when just given limited amount of leakage.

According to the ℓ-leakage-resilient anonymous property of IB-HPS, the advantage of any PPT adversary in Game 2 is negligible. Therefore the advantage of any PPT adversary in Game 0 is also negligible in κ, which concludes the Theorem 7.2. □

8 Instantiations of IB-HPS

IB-HPS is known to exist based on a variety of assumptions [ADN+10]. We first describe a construction of IB-HPS which is smooth but not anonymous, then describe five constructions of IB-HPS which are smooth and anonymous. We note that the last three constructions have been presented in [ADN+10]. For completeness, we interpret them using our generalized paradigm.

8.1 IB-HPS Based on the DBDH Assumption

We now describe an IB-HPS based on the DBDH assumption, which can be viewed as the backbone of the IBE scheme presented in [CDRW10].

Let $\mathbf{D}$ be a distribution distinguish problem based on the DBDH assumption. $\mathsf{SampDDP}(\kappa)$ runs $\mathsf{BLGroupGen}(\kappa)$ to generate $\text{PP} = (e, \mathbb{G}, \mathbb{G}_T, p)$, picks $g \xleftarrow{R} \mathbb{G}^*$, $u_0, u_1, \ldots, u_n \xleftarrow{R} \mathbb{G}$, $a, b, c, d \xleftarrow{R} \mathbb{Z}_p$, sets $mpk = (g, u_0, u_1, \ldots, u_n, e(g,g)^{ab}, e(g,g)^{cd})$, $msk = (g^{ab}, g^{cd})$; outputs an instance description $\Gamma = (X, W, PK, A, B, \mathsf{R})$ of $\mathbf{D}$, where $X = \mathbb{G} \times \mathbb{G} \times \mathbb{G}_T$, $W = \mathbb{Z}_p$, $PK = \mathbb{G}$, $\mathsf{R}_{pk} = \{(x, w) \in X \times W : ((pk^w, g^w, e(g,g)^{abw}), w)\}$, two collections of distributions A and B are specified by SampA and SampB below:

- $\mathsf{SampA}(pk)$: pick $w \xleftarrow{R} \mathbb{Z}_p$, output $x = (pk^w, g^w, e(g,g)^{abw}) \leftarrow A_{pk}$ and $w \in W$.
- $\mathsf{SampB}(pk)$: pick $w, w' \xleftarrow{R} \mathbb{Z}_p$, output $x = (pk^w, g^w, e(g,g)^{abw'}) \leftarrow B_{pk}$.

Let $\mathbf{H} = (\mathsf{H}, SK, PK, X, A, Y, \alpha)$ be a corresponding projective hash family, where $SK = \mathbb{Z}_p \times \mathbb{G} \times \mathbb{G}$, $Y = \mathbb{G}_T$. For $sk = (sk_1, sk_2, sk_3)$ and $x = (x_1, x_2, x_3)$, we define H as $\mathsf{H}_{sk}(x) = e(x_1, sk_3)e(x_2, sk_2)x_3^{sk_1}$.

Let $\mathbf{P}$ be an IB-HPS for $\mathbf{D}$ associating $\mathbf{H}$, which consists of four algorithms as below:

- $\mathsf{Setup}(\kappa)$: run $\mathsf{SampDDP}(\kappa)$ to generate $mpk = (g, u_0, u_1, \ldots, u_n, e(g,g)^{ab}, e(g,g)^{cd})$ and $msk = (g^{ab}, g^{cd})$; set the identity set I to be $\mathbb{Z}_p$, construct $\mathsf{IHF} : \mathbb{Z}_p \to \mathbb{G}$ as $\mathsf{IHF}(id) = u_0 \prod_{i=1}^{n} u_i^{id_i}$ (known as the Waters hash [Wat05]) where id_i denotes the i-th bit of identity id; $\sigma(msk, pk)$ is constructed as: parse msk as (msk_1, msk_2), pick $t, r \xleftarrow{R} \mathbb{Z}_p$, output $(t, msk_1 msk_2^{-t} pk^r, g^{-r})$.
- $\mathsf{KeyGen}(msk, id)$: compute $pk \leftarrow \mathsf{IHF}(id)$, output $sk \leftarrow \sigma(msk, pk)$.
- $\mathsf{Pub}(id, x, w)$: compute $pk \leftarrow \mathsf{IHF}(id)$, for $x = (pk^w, g^w, e(g,g)^{abw})$ output $y = e(g,g)^{cdw}$.
- $\mathsf{Priv}(sk, x)$: parse sk as (sk_1, sk_2, sk_3) and x as (x_1, x_2, x_3), output $y = e(x_1, sk_3)e(x_2, sk_2)x_3^{sk_1}$.

It is obvious that one can use the bilinear map as a tool to test if an invalid sample x is generated by SampB with respect to pk. Therefore the above IB-HPS is smooth but not anonymous. This provides us an evidence that for IB-HPS smoothness does not guarantee anonymity.

8.2 IB-HPS Based on the DSBDH Assumption

We now describe an IB-HPS based on the DSBDH assumption, which can be viewed as the backbone of the first IBE scheme presented in [Cor09].

Let $\mathbf{D}$ be a distribution distinguish problem based on the DSBDH assumption. $\mathsf{SampDDP}(\kappa)$ runs $\mathsf{BLGroupGen}(\kappa)$ to generate $\text{PP} = (e, \mathbb{G}, \mathbb{G}_T, p)$, picks $g \xleftarrow{R} \mathbb{G}^*$, $a \xleftarrow{R} \mathbb{Z}_p$, sets $mpk = (g, g_1 = g^a)$ and $msk = a$; outputs an instance description $\Gamma = (X, W, PK, A, B, \mathsf{R})$ of $\mathbf{D}$, where $X = \mathbb{G} \times \mathbb{G}_T$, $W = \mathbb{Z}_p$, $PK = \mathbb{G}$, $\mathsf{R}_{pk} = \{(x, w) \in X \times W : ((g^w, e(g_1, g_1)^w), w)\}$, two collections of distributions A and B are defined by SampA and SampB as below:

- $\mathsf{SampA}(pk)$: pick $w \xleftarrow{R} \mathbb{Z}_p$, output $x = (g^w, e(g_1, g_1)^w) \in X$ and $w \in W$.

- $\mathsf{SampB}(pk)$: pick $w, w' \xleftarrow{R} \mathbb{Z}_p$, output $x = (g^w, e(g_1, g_1)^{w'}) \in X$.

In this case, A_{pk}, B_{pk}, and R_{pk} are the same for all $pk \in PK$. We then write A, B, and R for simplicity.

Let $\mathbf{H} = (\mathsf{H}, SK, PK, X, A, Y, \alpha)$ be a corresponding projective hash family, where $SK = \mathbb{Z}_p \times \mathbb{G}$, $Y = \mathbb{G}_T$. For $sk = (sk_1, sk_2)$ and $x = (x_1, x_2)$, H is defined as $\mathsf{H}_{sk}(x) = e(x_1, sk_2)x_2^{sk_1}$.

Let $\mathbf{P}$ be an IB-HPS for $\mathbf{D}$ associating $\mathbf{H}$, which consists of four algorithms as below:

- $\mathsf{Setup}(\kappa)$: run $\mathsf{SampDDP}(\kappa)$ to generate $mpk = (g, g_1 = g^a)$ and $msk = a$. The identity set I is $\{0,1\}^*$ and IHF is a random oracle from $\{0,1\}^*$ to $\mathbb{G}$. $\sigma(msk, pk)$ is constructed as: pick $t \xleftarrow{R} \mathbb{Z}_p$, output $(t, (pk \cdot g_1^{-t})^{msk})$.
- $\mathsf{KeyGen}(msk, id)$: compute $pk \leftarrow \mathsf{IHF}(id)$, output $sk \leftarrow \sigma(msk, pk)$.
- $\mathsf{Pub}(id, x, w)$: compute $pk \leftarrow \mathsf{IHF}(id)$, for $x = (g^w, e(g_1, g_1)^w)$ output $y = e(pk, g_1)^w$.
- $\mathsf{Priv}(sk, x)$: parse sk as (sk_1, sk_2) and x as (x_1, x_2), output $y = e(x_1, sk_2)x_2^{sk_1}$.

The above IB-HPS is smooth and anonymous.

8.3 IB-HPS Based on the DBDH Assumption

We now describe an IB-HPS based on the DBDH assumption, which can be viewed as the backbone of the second IBE scheme presented in [Cor09].

Let $\mathbf{D}$ be a distribution distinguish problem based on the DBDH assumption. $\mathsf{SampDDP}(\kappa)$ runs $\mathsf{BLGroupGen}(\kappa)$ to generate $\mathrm{PP} = (e, \mathbb{G}, \mathbb{G}_T, p)$, picks $g \xleftarrow{R} \mathbb{G}^*$, $a \xleftarrow{R} \mathbb{Z}_p$, $g_2 \xleftarrow{R} \mathbb{G}$, sets $mpk = (g, g_1 = g^a, g_2)$, $msk = a$; outputs an instance description $\Gamma = (X, W, PK, A, B, \mathsf{R})$ of $\mathbf{D}$, where $X = \mathbb{G} \times \mathbb{G}_T$, $W = \mathbb{Z}_p$, $PK = \mathbb{G}$, $\mathsf{R}_{pk} = \{(x, w) \in X \times W : ((g^w, e(g_1, g_2)^w), w)\}$, two collections of distributions A and B are defined by SampA and SampB as below:

- $\mathsf{SampA}(pk)$: pick $w \xleftarrow{R} \mathbb{Z}_p$, output $x = (g^w, e(g_1, g_2)^w) \leftarrow A_{pk}$ and $w \in W$.
- $\mathsf{SampB}(pk)$: pick $w, w' \xleftarrow{R} \mathbb{Z}_p$, output $x = (g^w, e(g_1, g_2)^{w'}) \in B_{pk}$.

In this case, A_{pk}, B_{pk}, and R_{pk} are the same for all $pk \in PK$. We then write A, B, and R for simplicity.

Let $\mathbf{H} = (\mathsf{H}, SK, PK, X, A, Y, \alpha)$ be a corresponding projective hash family, where $SK = \mathbb{Z}_p \times \mathbb{G}$, $Y = \mathbb{G}_T$. For $sk = (sk_1, sk_2)$ and $x = (x_1, x_2)$, H is defined as $\mathsf{H}_{sk}(x) = e(x_1, sk_2)x_2^{sk_1}$.

Let $\mathbf{P}$ be an IB-HPS for $\mathbf{D}$ associating $\mathbf{H}$, which consists of four algorithms as below:

- $\mathsf{Setup}(\kappa)$: run $\mathsf{SampDDP}(\kappa)$ to generate $mpk = (g, g_1 = g^a, g_2)$, $msk = a$; set the identity set I to be $\{0,1\}^*$, model IHF as a random oracle from $\{0,1\}^*$ to $\mathbb{G}$. $\sigma(msk, pk)$ is constructed as: pick $t \xleftarrow{R} \mathbb{Z}_p$, output $(t, (pk \cdot g_2^{-t})^{msk})$.
- $\mathsf{KeyGen}(msk, id)$: compute $pk \leftarrow \mathsf{IHF}(id)$, output $sk \leftarrow \sigma(msk, pk)$.
- $\mathsf{Pub}(id, x, w)$: compute $pk \leftarrow \mathsf{IHF}(id)$, for $x = (g^w, e(g_1, g_2)^w)$ output $y = e(pk, g_1)^w$.
- $\mathsf{Priv}(sk, x)$: parse sk as (sk_1, sk_2) and x as (x_1, x_2), output $y = e(x_1, sk_2)x_2^{sk_1}$.

As shown in [Cor09], the above IB-HPS is smooth and anonymous.

8.4 IB-HPS Based on the DTABDHE Assumption

We now describe an IB-HPS based on the DTABDHE assumption [Gen06], which can be viewed as the backbone of Gentry's IBE [Gen06].

Let $\mathbf{D}$ be a distribution distinguish problem based on the DTABDHE assumption. $\mathsf{SampDDP}(\kappa)$ runs $\mathsf{BLGroupGen}(\kappa)$ to generate $\mathrm{PP} = (e, \mathbb{G}, \mathbb{G}_T, p)$, picks $g, h \leftarrow \mathbb{G}^*$ and $a \xleftarrow{R} \mathbb{Z}_p$, sets $mpk = (g, g_1 = g^a, h)$ and $msk = a$; outputs an instance description $\Gamma = (X, W, PK, A, B, \mathsf{R})$ of $\mathbf{D}$, where $X = \mathbb{G} \times \mathbb{G}_T$, $W = \mathbb{Z}_p$, $PK = \mathbb{Z}_p$, $\mathsf{R}_{pk} = \{(x, w) \in X \times W : ((g_1^w g^{-w \cdot pk}, e(g, g)^w), w)\}$, two collections of distributions A and B are defined by SampA and SampB as below:

- $\mathsf{SampA}(pk)$: pick $w \xleftarrow{R} \mathbb{Z}_p$, output $x = (g_1^w g^{-w \cdot pk}, e(g,g)^w) \leftarrow A_{pk}$ and $w \in W$.
- $\mathsf{SampB}(pk)$: pick $w, w' \xleftarrow{R} \mathbb{Z}_p$, output $x = (g_1^w g^{-w \cdot pk}, e(g,g)^{w'}) \leftarrow B_{pk}$.

Let $\mathbf{H} = (\mathsf{H}, SK, PK, X, A, Y, \alpha)$ be a corresponding projective hash family, where $SK = \mathbb{Z}_p \times \mathbb{G}$, $Y = \mathbb{G}_T$. For $sk = (sk_1, sk_2)$ and $x = (x_1, x_2)$, H is defined as $\mathsf{H}_{sk}(x) = e(x_1, sk_2) x_2^{sk_1}$.

Let $\mathbf{P}$ be an IB-HPS for $\mathbf{D}$ associating $\mathbf{H}$, which consists of four algorithms as below:

- $\mathsf{Setup}(\kappa)$: run $\mathsf{SampDDP}(\kappa)$ to generate $mpk = (g, g_1 = g^a, h)$ and $msk = a$; set the identity set I to be $\mathbb{Z}_p$, IHF is an identity function. $\sigma(msk, pk)$ is constructed as: pick $t \in \mathbb{Z}_p$, output $(t, (hg^{-t})^{1/(msk - pk)})$.
- $\mathsf{KeyGen}(msk, id)$: set $pk = id$, output $\sigma(msk, pk)$.
- $\mathsf{Pub}(id, x, w)$: set $pk = id$, for $x = (g_1^w g^{-w \cdot pk}, e(g,g)^w)$ output $y = e(g,h)^{-w}$.
- $\mathsf{Priv}(sk, x)$: parse $sk = (sk_1, sk_2)$ and $x = (x_1, x_2)$, output $y = e(x_1, sk_2) x_2^{sk_1}$.

As shown in [Gen06], the above IB-HPS is smooth and anonymous.

8.5 IB-HPS Based on the QR Assumption

We now describe an IB-HPS based on the QR assumption, which can be viewed as the backbone of the IBE scheme presented in [BGH07].

Let $\mathbf{D}$ be a distribution distinguish problem based on the QR assumption. $\mathsf{SampDDP}(\kappa)$ runs $\mathsf{RSAGen}(\kappa)$ to generate two primes p and q, sets $mpk = (N, u, \mathcal{Q})$ (where $N = pq$ and $u \xleftarrow{R} J(N) \backslash QR(N)$ and $\mathcal{Q}$ is the algorithm defined in [ADN+10, Appendix C]) and $msk = (p, q)$, outputs an instance description $\Gamma = (X, W, PK, A, B, \mathsf{R})$ of $\mathbf{D}$, where $X = J(N) \times \{\pm 1\}$, $W = \mathbb{Z}_N$, $PK = J(N)$, $\mathsf{R}_{pk} = \{(x, w) \in X \times W : ((w^2, (\frac{\tau(w)}{N})), w)\}$, two collections of distributions A and B are specified by SampA and SampB as below:

- $\mathsf{SampA}(pk)$: pick $w \xleftarrow{R} \mathbb{Z}_p$, set $x_1 = w^2$, run $\mathcal{Q}(N, u, 1, x_1)$ to obtain τ and compute $x_2 = (\frac{\tau(w)}{N})$, output $x = (x_1, x_2) \in X$ and $w \in W$.
- $\mathsf{SampB}(pk)$: pick $x_1 \xleftarrow{R} J(N) \backslash QR(N)$, $x_2 \xleftarrow{R} \{\pm 1\}$, output $x = (x_1, x_2) \in X \backslash L$.

In this case, A_{pk}, B_{pk}, and R_{pk} are the same for all $pk \in PK$. We then write A, B, and R for simplicity.

Let $\mathbf{H} = (\mathsf{H}, SK, PK, X, A, Y, \alpha)$ be a corresponding projective hash family, where $SK = \mathbb{Z}_N$, $Y = \{\pm 1\}$. For $sk = r$ and $x = (x_1, x_2)$, $\mathsf{H}_{sk}(x) = y$ is defined as when $r^2 = pk$ output $(\frac{f(r)}{N})$ else output $x_2(\frac{\bar{f}(r)}{N})$, where f, $\bar{f}$ are the polynomials output by $\mathcal{Q}(N, u, R, x_1)$.

Let $\mathbf{P}$ be an IB-HPS for $\mathbf{D}$ associating $\mathbf{H}$, which consists of four algorithms as below:

- $\mathsf{Setup}(\kappa)$: run $\mathsf{SampDDP}(\kappa)$ to generate $mpk = (N = pq, u, \mathcal{Q})$ and $msk = (p, q)$; set the identity set I to be $\{0,1\}^*$ and IHF is a random oracle from $\{0,1\}^*$ to $J(N)$. $\sigma(msk, pk)$ is constructed as: let $a \in \{0, 1\}$ be the unique choice for which $u^a pk \in QR(N)$, let $\{r_1, r_2, r_3, r_4\}$ be the four square-roots of $u^a pk$ so that $r_1 < r_2 < r_3 < r_4$ (in $\mathbb{Z}_N$) and $r_1 = -r_4$, $r_2 = -r_3$, output $r \xleftarrow{R} \{r_1, r_2\}$.
- $\mathsf{KeyGen}(msk, id)$: compute $pk = \mathsf{IHF}(id)$, output $sk = \alpha(msk, pk)$.
- $\mathsf{Pub}(id, x, w)$: compute $pk = \mathsf{IHF}(id)$, for $x = (x_1, x_2)$, run $\mathcal{Q}(N, u, pk, x_1)$ to obtain a polynomial g, output $y = (\frac{g(w)}{N})$.
- $\mathsf{Priv}(sk, x)$: suppose $sk = r$ for id and $x = (x_1, x_2)$, compute $pk = \mathsf{IHF}(id)$, run $\mathcal{Q}(N, u, pk, x_1)$ to obtain polynomials f, $\bar{f}$. If $r^2 = pk$ output $y = (\frac{f(r)}{N})$, else output $y = x_2(\frac{\bar{f}(r)}{N})$.

As shown in [BGH07], the above IB-HPS is smooth and anonymous.

8.6 IB-HPS Based on the DLWE Assumption

We now describe an IB-HPS based on the DLWE assumption, which can be viewed as the backbone of the IBE scheme presented in [GPV08].

The distribution distinguish problem $\mathbf{D}$ and the projective hash family $\mathbf{H}$ are the same as that in HPS defined in subsection 5.1.

Let $\mathbf{P}$ be an IB-HPS for $\mathbf{D}$ associating $\mathbf{H}$, which consists of four algorithms as below:

$\mathsf{Setup}(\kappa)$: run $\mathsf{SampDDP}(\kappa)$ to generate $mpk = (\mathbb{A}, f_{\mathbb{A}})$ and $msk = \mathbb{S}$. The identity set I is $\{0,1\}^*$ and the identity mapping function IHF is a random oracle from $\{0,1\}^*$ to $\mathbb{Z}_q^n$. The inversion of α is constructed as $\sigma(msk, pk) = f_{\mathbb{A}}^{-1}(pk)$ using the preimage sampler with $msk = \mathbb{S}$.
$\mathsf{KeyGen}(msk, id)$: compute $pk \leftarrow \mathsf{IHF}(id)$, output $sk \leftarrow \sigma(msk, pk)$.
$\mathsf{Priv}(sk, x)$: parse $x = (p, v)$, if $|v - sk^T p| \leq \frac{q-1}{4}$ then output $y = 1$ else output $y = 0$.
$\mathsf{Pub}(id, x, w)$: compute $pk \leftarrow \mathsf{IHF}(id)$, parse $x = (p, v)$, if $|v - pk^T w| \leq \frac{q-1}{4}$ then set $y = 1$ else set $y = 0$.

As shown in [GPV08], the above IB-HPS is smooth and anonymous.

9 Leakage-Resilient Public-Key Encryption with Keyword Search

Public-key encryption with keyword search (PEKS) [BCOP04] is a useful primitive to provide the functionality of "searching on encrypted data" for public-key encryption (PKE). It allows one to delegate his searching ability on encrypted data to a third party without impacting the data privacy and the keyword privacy.

A PEKS scheme [BCOP04] consists of four PPT algorithms as below:

- $\mathsf{Setup}(\kappa)$: take as input a security parameter κ, output a public/secret key pair (pk_A, sk_A). Let the keyword space be W, the token space be T.
- $\mathsf{PEKS}(pk_A, w)$: take as input a public key pk_A and a keyword w, output a searchable encryption s.
- $\mathsf{TokenGen}(sk_A, w)$: take as input a secret key sk_A and a keyword w, output a token t_w.
- $\mathsf{Test}(t_w, s)$: take as input a public key pk, a searchable encryption s, output a bit $b \in \{0,1\}$.

The correctness of PEKS requires that for any $w \in W$, we have:

$$\Pr[\mathsf{Test}(\mathsf{TokenGen}(sk_A, w), \mathsf{PEKS}(pk_A, w)) = 1] \geq 1 - \mathsf{negl}(\kappa)$$

where the probability is taken over the random coins used by Setup, PEKS, and $\mathsf{ToeknGen}$. On the other hand, the consistency of PEKS requires that Test outputs 0 with overwhelmingly probability if s is not an encryption of w.

9.1 Leakage Model for PEKS

Intuitively, a PEKS scheme is said to be secure if no PPT adversary cannot distinguish the keyword under which a ciphertext was generated. Boneh *et al.* [BCOP04] formally defined the security of PKES under chosen-plaintext attack in the traditional sence. However, there is no work considering the security of PEKS in the presence of leakage.

In what follows, we define the leakage-resilient security of PEKS by modifying the usual security game of PEKS (against chosen-plaintext attack) appropriately in the bounded-leakage model. Informally speaking, a PEKS scheme is leakage-resilient secure if it retains security even an adversary can obtain partial information about the tokens of the keywords.

We consider the security for PEKS against chosen-plaintext attack in the presence of token leakage. Advantage of an adversary $\mathcal{A}$ is defined as:

$$\mathrm{Adv}_{\mathcal{A}}(\kappa) = \Pr\left[b = b' : \begin{array}{l} (pk_A, sk_A) \leftarrow \mathsf{Setup}(\kappa); \\ (w_0, w_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{reveal}}(\cdot), \mathcal{O}_{\text{leak}}(\cdot,\cdot)}(pk_A); \\ b \xleftarrow{R} \{0,1\}; \\ s^* \leftarrow \mathsf{Encrypt}(w_b, m); \\ b' \leftarrow \mathcal{A}^{\mathcal{O}_{\text{reveal}}(\cdot), \mathcal{O}_{\text{leak}}(\cdot,\cdot)}(s^*) \end{array} \right] - \frac{1}{2}$$

where $\mathcal{O}_{\text{reveal}}(\cdot)$ is an oracle that on input w, returns $t_w \leftarrow \mathsf{TokenGen}(sk_A, w)$, $\mathcal{O}_{\text{leak}}(\cdot,\cdot)$ is an oracle that on input w and a function $f : T \to \{0,1\}^*$, returns $f(t_w)$. The restrictions are that the reveal oracle is only available for keywords other than w_0, w_1, and after seeing the challenge ciphertext s^* the leak oracle is only available for keywords other than w_0, w_1. We say $\mathcal{A}$ an ℓ-leakage adversary if the sum of output length of all functions that it submits to the leakage oracle $\mathcal{O}_{\text{leak}}(w, \cdot)$ for any single $w \in W$ is less than ℓ. An PEKS scheme is said to be ℓ-leakage-resilient IND-CPA secure if for any PPT ℓ-leakage adversary $\mathcal{A}$, its advantage defined as above is negligible in κ.

9.2 Generic Construction of Leakage-Resilient PEKS

We first review the generic transform due to [BCOP04, ABC+05] from anonymous IBE to PEKS, then show that this transform also provides us a generic method to compile a leakage-resilient anonymous IBE scheme into a leakage-resilient secure PEKS scheme.

Boneh *et al.* [BCOP04] presented a transform from an IBE scheme to a PEKS scheme (knows as the BDOP transform). Subsequently, Abdalla *et al.* [ABC+05] revised the BDOP transform and gave a formal proof for it. The resulting randomized BDOP transform is refereed to as the new BDOP transform. Starting from an IBE scheme (Setup, KeyGen, $\mathsf{Encrypt}$, $\mathsf{Decrypt}$), the new BDOP transform creates a PEKS scheme (Setup, PEKS, $\mathsf{TokenGen}$, Test) as follows:

- PEKS.$\mathsf{Setup}(\kappa)$: run $(mpk, msk) \leftarrow$ IBE.$\mathsf{Setup}(\kappa)$, set $pk_A = mpk$, $sk_A = msk$, return the key pair (pk_A, sk_A).
- PEKS.$\mathsf{Encrypt}(pk_A, w)$: pick $s_1 \xleftarrow{R} M$ (here M is message set of the underlying IBE scheme), compute $s_2 \leftarrow$ IBE.$\mathsf{Encrypt}(w, s_1)$ where w serves as the identity, return the searchable encryption $s = (s_1, s_2)$.
- PEKS.$\mathsf{TokenGen}(sk_A, w)$: run IBE.$\mathsf{KeyGen}(sk_A, w)$ to obtain a private key sk_w for "identity" w, return $t_w = sk_w$.
- PEKS.$\mathsf{Test}(pk_A, s, t_w)$: parse $s = (s_1, s_2)$, decide if $s_1 =$ IBE.$\mathsf{Decrypt}(s_2, t_w)$. If so, return 1. Otherwise, return 0.

Leakage-Resilient PEKS from Leakage-Resilient IBE. Be aware of the close resemblance between the (leakage-resilient) security game for PEKS and (leakage-resilient) anonymity game for IBE, leakage-resilient anonymous IBE is likely to imply leakage-resilient secure PEKS. We formally capture this intuition by the following theorem.

Theorem 9.1 *For PEKS scheme constructed from IBE scheme via the new BDOP transform, if the IBE scheme is ℓ-leakage-resilient ANO-CPA anonymous, then the resulting PEKS scheme is ℓ-leakage-resilient IND-CPA secure.*

Proof. The proof of this theorem is rather straightforward. Suppose $\mathcal{A}$ is an ℓ-leakage adversary against the security of the PEKS scheme, we can create an ℓ-leakage adversary $\mathcal{B}$ against the anonymity of the underlying IBE scheme. $\mathcal{B}$ interacts with its own challenger $\mathcal{CH}$ in a leakage-resilient anonymity game of IBE, and at the same time plays the role of a challenger for $\mathcal{A}$ in a leakage-resilient security game of PEKS as below.

Setup: $\mathcal{B}$ is given mpk of the underlying IBE scheme. $\mathcal{B}$ forwards it to $\mathcal{A}$ as the public key pk_A.
Phase 1: Upon receiving the token reveal queries and test leakage queries issued by $\mathcal{A}$, $\mathcal{B}$ responds as follows:

- Token reveal query $\langle w \rangle$: $\mathcal{B}$ issues the private key reveal query for "identity" w to its own challenger $\mathcal{CH}$ and forwards the reply to $\mathcal{A}$.
- Token leakage query $\langle w, f_i \rangle$: $\mathcal{B}$ issues the private key leak query $\langle w, f_i \rangle$ to its own challenger $\mathcal{CH}$ and forwards the reply to $\mathcal{A}$.

Challenge: When $\mathcal{A}$ outputs two keywords w_0 and w_1 where it wants to be challenged on, $\mathcal{B}$ picks $m \xleftarrow{R} M$, then sends it with w_0 and w_1 (serve as two target identities) to $\mathcal{CH}$. When $\mathcal{B}$ obtains the challenge ciphertext $c = \text{IBE}.\mathsf{Encrypt}(mpk, w_b, m)$, it sends $s = (c, m)$ to $\mathcal{A}$ as the challenge.
Phase 2: $\mathcal{A}$ may issue more token reveal and token leakage queries with the restriction that they are not related to either w_0 or w_1. $\mathcal{B}$ responds the same way as in Phase 1.
Guess: As soon as $\mathcal{A}$ outputs its guess b' for b, $\mathcal{B}$ outputs b' to its own challenger.

It is easy to see that $\mathcal{B}$ provides a perfect simulation for $\mathcal{A}$, so $\mathcal{B}$ has the same advantage against the ℓ-leakage anonymity of the underlying IBE scheme as $\mathcal{A}$ wins the above game. Theorem 9.1 immediately follows. □

Remark 2. The leakage model for PEKS defined in subsection 9.1 only allows leakage on one token for each keyword. It can be further strengthened to allow leakage on several tokens for each keyword as well as the user's private key. Notice the correspondence between PEKS and IBE indicated by the new BDOP transform, one can immediately construct a PEKS scheme that can tolerate leakage on several tokens per keyword as well as the user's private key from an anonymous IBE scheme that allows leakage on several private keys for each identity as well as the master secret key. Recently, Lewko *et al.* [LRW11] presented the first IBE scheme which can tolerate leakage leakage on several keys for each identity as well as leakage on the master secret key from dual system encryption. However, it is not anonymous.

10 Concluding Remarks

The role of master secret key. Firstly, we highlight some subtle details in our generalization of (IB)-HPS. In the definition of HPS and IB-HPS, we make explicit the master secret key msk, though it is not used in the algorithms of HPS. From narrative aspect, this treatment allows us to describe HPS and IB-HPS in a unified manner. From technical aspect, this treatment allows us to define (leakage-resilient) smoothness and anonymity for (IB)-HPS as strong as possible. In fact, we requires (leakage-resilient) smoothness and anonymity hold even the adversary can obtain the entire information of msk. Such strengthened definitions have not been explicitly addressed prior to this work. We emphasize that our enhancement is safe and free, in that the smoothness is acquired via the redundancy of secret key while the anonymity is related to the property of algorithm SampB. It is also interesting to note that when reducing the indistinguishability of (IB)-HPS to the underlying assumptions, the hard instance is usually embedded into msk. This explains the difficulty of constructing IBE that allows leakage on msk via the methodology of hash proof system.

The relationship between smoothness and anonymity. In the cases of HPS and IB-HPS, when the algorithm SampB is independent of pk, smoothness instantly implies anonymity, in that the sample x does not contain any information of pk ($\mathsf{IHF}(id)$) and the corresponding hashing value y is pseudo-random in Y. This observation provides us a simple approach to attain anonymity for many encryption schemes.[4] The instantiations presented in subsections 5.1, 8.2 8.3, 8.5, 8.6 exactly follow this approach. However, there do exist anonymous (IB)-HPS falling outside this approach, for instance, the IB-HPS underlying the Gentry IBE [Gen06] does not adapt this approach but is anonymous, as we showed in subsection 8.4. We also stress that smoothness does not always guarantee anonymity, for example, the IB-HPS that underlying the IBE scheme [CDRW10] is smooth but not anonymous, as we show in subsection 8.1.

Chosen-ciphertext security. In the main body of this paper, we restrict our attention to (leakage-resilient) security against only chosen-plaintext attacks. Technically, there are two generic method to bootstrap the security to the setting of (adaptive) chosen-ciphertext attacks. As indicated in [NS09], Naor-Yung "double encryption" [DDN00] can also be used as a generic transform from CPA-security to CCA-security in the presence of key-leakage. However, this approach is only of theoretical interest in that the generic non-interactive zero-knowledge proofs are generally inefficient. From the practical aspect, we note that Cramer and Shoup [CS02] already presented a direct and elegant construction of CCA-secure PKE from HPS. In more details, except a smooth PHF $\mathbf{H}$ from X to Y (which is sufficient to yield CPA-secure PKE), a corresponding PHF $\tilde{\mathbf{H}}$ from $X \times M$ to $\tilde{Y}$ satisfying universal$_2$ property is also needed. One then can obtain a CCA-secure PKE by using $\mathbf{H}$ and $\tilde{\mathbf{H}}$ to perform "double encryption"-like operation. We observe that universal$_2$ (see definition in [CS02]) is a much stronger notion than smoothness. Intuitively, for the distribution of $y = \mathsf{H}_{sk}(x)$ conditioned on some fixed $pk = \alpha(sk)$, universal property requires that for *every* point $x \in X \backslash L$ the distribution is close to the uniform, while smooth property just requires that for *almost every* point $x \in X \backslash L$ the distribution is close to the uniform. It is also interesting to realize that the compatibility between projection and universality in the original HPS comes from the fact that L and $X \backslash L$ are disjointed. In the context of our generalization, we do not require the partition on X. Alternatively, we re-define the projection and smoothness with respect to distributions A and B over X, and $|A|$ and $|B|$ are not necessarily to be disjointed. Therefore, the projection and universality contradict with each other when the supports of A and B are identical (e.g. the case of lattice-based HPS shown in subsection 5.1). However, it might still be possible to construct a HPS satisfying approximate projection and universality simultaneously when $|A| \cap |B|$ is not empty (the situation that the action of H_{sk} on $|A| \cap |B|$ is completely undetermined, and $\Pr[x \in |A| \cap |B| : x \leftarrow A]$ is negligible.). We left concrete construction of such generalized (IB)-HPS as an open problem.

Extensions. Our generalization idea extends naturally to many variants of (IB)-HPS. Most notably, in the dual HPS introduced in [Wee12], the set Π_{Y} (corresponding to YES instances) and Π_{N} (corresponding to NO instances) are required to be disjointed, and the indistinguishability is respect to two uniform distributions over them. However, such restrictions makes the instantiation from LWE presented in [Wee12, Section 8] do not fit the original paradigm of dual HPS exactly. Similar to the case of (IB)-HPS, these restrictions are not necessary in essence, it is thus reasonable to generalize dual HPS by centring around distribution distinguish problem instead of subset membership problem. Interestingly, the generalized dual HPS immediately implies *almost-always* LTDFs, which is a slightly relaxed definition of LTDFs suggested by Peikert and Waters [PW08].

References

[ABC+05] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-lee, Gregory Neven, Pascal, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. In *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *LNCS*, pages 205–222, 2005.

[4] The "anonymous encapsulation" property proposed in [ADN+10] exactly embodies the same idea.

[ACP09] Michel Abdalla, Céline Chevalier, and David Pointcheval. Smooth projective hashing for conditionally extractable commitments. In *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *LNCS*, pages 671–689. Springer, 2009.

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

[ADW09] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *LNCS*, pages 36–54. Springer, 2009.

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

[AR02] Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *J. Cryptology*, 15(2):103–127, 2002.

[BBDP01] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 566–582. Springer, 2001.

[BBO07] Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. In *Advances in Cryptology - CRYPTO 2007*, volume 4622 of *LNCS*, pages 535–552. Springer, 2007.

[BCOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *Advances in Cryptology - EUROCRYPT 2004*, volume 3621 of *LNCS*, pages 506–522, 2004.

[BF03] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. *SIAM Journal on Computation*, 32:586–615, 2003.

[BFOR08] Mihir Bellare, Marc Fischlin, Adam O'Neill, and Thomas Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In *Advances in Cryptology - CRYPTO 2008*, volume 5157 of *LNCS*, pages 360–378. Springer, 2008.

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

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *Advances in Cryptology - EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, 2009.

[BPV12] Olivier Blazy, David Pointcheval, and Damien Vergnaud. Round-optimal privacy-preserving protocols with smooth projective hash functions. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012*, volume 7194 of *LNCS*, pages 94–111. Springer, 2012.

[BS11] Zvika Brakerski and Gil Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. In *Advances in Cryptology - CRYPTO 2011*, volume 6841 of *LNCS*, pages 543–560. Springer, 2011.

[CDH$^+$00] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-resilient functions and all-or-nothing transforms. In *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 453–469. Springer, 2000.

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

[CLC11] Yu Chen, Song Luo, and Zhong Chen. A new leakage-resilient ibe scheme in the relative leakage model. In *Data and Applications Security and Privacy XXV - 25th Annual IFIP WG 11.3 Conference, DBSec 2011*, volume 6818 of *LNCS*, pages 263–270. Springer, 2011.

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

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Advances in Cryptology - CRYPTO 1998*, volume 1462 of *LNCS*, pages 13–25, 1998.

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

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.

[Des00] Anand Desai. The security of all-or-nothing encryption: Protecting against exhaustive key search. In *Advances in Cryptology - CRYPTO 2000*, volume 1880 of *LNCS*, pages 359–375. Springer, 2000.

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

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008*, pages 293–302. IEEE Computer Society, 2008.

[Fis99] Marc Fischlin. Pseudorandom function tribe ensembles based on one-way permutations: Improvements and applications. In *Advances in Cryptology - EUROCRYPT 1999*, volume 1592 of *LNCS*, pages 432–445. Springer, 1999.

[FKPR10] Sebastian Faust, Eike Kiltz, Krzysztof Pietrzak, and Guy N. Rothblum. Leakage-resilient signatures. In *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010*, volume 5978 of *LNCS*, pages 343–360. Springer, 2010.

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

[GL06] Rosario Gennaro and Yehuda Lindell. A framework for password-based authenticated key exchange[1]. *ACM Trans. Inf. Syst. Secur.*, 9(2):181–234, 2006.

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

[HK12] Shai Halevi and Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. *J. Cryptology*, 25(1):158–193, 2012.

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

[HO12] Brett Hemenway and Rafail Ostrovsky. On homomorphic encryption and chosen-ciphertext security. In *Public Key Cryptography - PKC 2012*, volume 7293 of *LNCS*, pages 52–65. Springer, 2012.

[Kil07] Eike Kiltz. Chosen-ciphertext secure key-encapsulation based on gap hashed diffie-hellman. In *Public Key Cryptography - PKC 2007*, volume 4450 of *LNCS*, pages 282–297. Springer, 2007.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. Smooth projective hashing and password-based authenticated key exchange from lattices. In *Advances in Cryptology - ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 636–652. Springer, 2009.

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

[LY12] Benoît Libert and Moti Yung. Non-interactive cca-secure threshold cryptosystems with adaptive security: New framework and constructions. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012*, volume 7194 of *LNCS*, pages 75–93. Springer, 2012.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, volume 2951 of *LNCS*, pages 278–296. Springer, 2004.

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

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *STOC 2008*, pages 187–196, 2008.

[Sha02] Ronen Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, 77:67–95, 2002.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127, 2005.

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636, 2009.

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

[ZHI07] Rui Zhang, Goichiro Hanaoka, and Hideki Imai. Orthogonality between key privacy and data privacy, revisited. In *Information Security and Cryptology, Third SKLOIS Conference, Inscrypt 2007*, volume 4990 of *LNCS*, pages 313–327. Springer, 2007.