All-but-One Dual Projective Hashing and Its Applications

Zongyang Zhang[1,4], Yu Chen[2,⋆], Sherman S.M. Chow[3], Goichiro Hanaoka[1], Zhenfu Cao[4], and Yunlei Zhao[5]

[1] National Institute of Advanced Industrial Science and Technology (AIST), Japan
[2] State Key Laboratory of Information Security (SKLOIS), Institute of Information Engineering, Chinese Academy of Sciences, China
[3] Department of Information Engineering, The Chinese University of Hong Kong
[4] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[5] Software School, Fudan University, China
zongyang.zhang@aist.go.jp, chenyu@iie.ac.cn, sherman@ie.cuhk.edu.hk, hanaoka-goichiro@aist.go.jp, zfcao@sjtu.edu.cn, ylzhao@fudan.edu.cn

Abstract. Recently, Wee (EUROCRYPT'12) introduced the notion of dual projective hashing as an extension of the Cramer-Shoup projective hashing, with a simple construction of lossy trapdoor functions, and a simple construction of deterministic encryption schemes which is chosen-plaintext-attack secure with respect to hard-to-invert auxiliary input. In this work, we further extend it to the all-but-one setting by introducing the notion of all-but-one dual projective hashing.

- We provide a simple construction of all-but-one lossy trapdoor functions. Our construction encompasses many known constructions of all-but-one lossy trapdoor functions, as presented by Peikert and Waters (STOC'08), and Freeman et al. (JoC'13). Particularly, we present a new construction of all-but-one lossy trapdoor functions based on the DLIN assumption, which can be viewed as an extension of Freeman et al.'s DDH-based construction to the DLIN setting, and therefore solves an open problem left by Freeman et al.
- We also provide a general construction of chosen-ciphertext-attack (CCA) secure deterministic encryption schemes in the standard model, under an additional assumption about the projective map. This extends the general approach of designing CCA secure deterministic encryption schemes by Boldyreva, Fehr and O'Neill (CRYPTO'08). In addition, we present a new construction of CCA secure deterministic encryption schemes based on the DLIN assumption.

Keywords: Smooth projective hashing, ABO lossy trapdoor function, deterministic encryption, CCA security.

1 Introduction

In 1998, Cramer and Shoup [9] presented the first efficient public key encryption scheme which is chosen-ciphertext-attack (CCA) secure in the standard model,

⋆ Corresponding author.

I. Boureanu, P. Owesarski, and S. Vaudenay (Eds.): ACNS 2014, LNCS 8479, pp. 181–198, 2014.

under the decisional Diffie-Hellman assumption. Towards a general paradigm of constructing CCA secure public key encryption schemes, they [10] abstracted the above work to hash proof system (HPS). At the heart of HPS lies a primitive dubbed "smooth projective hashing". Thereafter, the smooth projective hashing and its variants have found numerous applications beyond CCA security, including password-based authenticated key exchange [15,19], extractable commitment [1], lossy encryption [5], leakage-resilient public key encryption [21], privacy-preserving interactive protocols [6], oblivious transfer [17], etc.

Informally, a smooth projective hashing is a family of keyed hash functions $\{\mathsf{H}_k\}$ whose input u is from some hard language (consisting of YES instances and NO instances). There are two ways to compute the function. First, knowing the hashing key k, one can compute the hash function on every instances in its domain. Second, knowing a projective key $\alpha(k)$ where α is a projective map, one can compute the hash function for each YES instance as long as it additionally knows the associated "witness". This means that the hash value $\mathsf{H}_k(u)$ is completely determined by $\alpha(k)$ and u, and this is therefore called the *projective* property. The other property, *smoothness*, means that the projective key $\alpha(k)$ gives (almost) no information about the value of the hash function on NO instance, i.e., the value of the hash function is completely undetermined.

Regarding evaluation on NO instances, instead of smoothness, Wee [25] considered *invertibility* that, for any NO instance u, one can compute the hashing key k given the projective key $\alpha(k)$ and the hash value $\mathsf{H}_k(u)$ together with an inversion trapdoor of u. This alternative introduced the notion of dual projective hashing (DPH), where "dual" means that roles of u and k are exchanged. This is why it is more convenient to write the function as $\Lambda_u(k) := \mathsf{H}_k(u)$. Moreover, in typical applications of smooth projective hashing, YES instances are used for functionality/correctness and NO instances are used to establish security. In contrast, in applications of dual projective hashing, YES instances are used to establish security, and NO instances are used for functionality/correctness.

Wee [25] showed a simple construction of lossy trapdoor functions via dual projective hashing and presented instantiations of dual projective hashing from Diffie-Hellman assumptions like Decisional Diffie-Hellman (DDH) and and Decisional Linear (DLIN), number-theoretic assumptions like Quadratic Residuosity (QR) and Decisional Composite Residuosity (DCR), and lattice-based assumptions like Learning-with-Error (LWE). It unifies (with slight changes) almost all known constructions of lossy trapdoor functions in [22,13]. When considering chosen-ciphertext security for encryption, many constructions based on lossy-trapdoor function rely on a more generalized all-but-one (ABO) lossy trapdoor functions [22]. It is natural to ask whether we can find an abstraction framework to unify existing ABO lossy trapdoor functions.

Dual projective hashing also leads to a simple construction of deterministic encryption scheme (with respect to hard-to-invert auxiliary input) [25]. Since it only achieves chosen-plaintext security, it is natural to ask whether we can achieve CCA security using dual projective hashing, or if we can get another general framework for CCA secure deterministic encryption schemes.

1.1 Overview of Our Results

We introduce the notion of ABO dual projective hashing. We consider a family of projective hash functions $\{\mathsf{H}_k\}$ indexed by a hashing key k and whose inputs are (tag, u). Here, we do not consider YES or NO instances. For any initial parameter $tag^* \in \text{TAG}$, and any u generated by some efficient algorithm together with tag^*, if $tag = tag^*$, we require *projective* property that the hash value $\mathsf{H}_k(tag, u)$ is completely determined by u and $\alpha(k)$; otherwise we require *invertibility* that there is some inversion trapdoor allowing us to efficiently recover k given $(\alpha(k), \mathsf{H}_k(tag, u))$ along with u. In addition, we require the *hidden projective tag* property that a randomly chosen input u under any $tag \in \text{TAG}$ is computationally indistinguishable from a randomly chosen input u' under another different tag $tag' \in \text{TAG}$. When $\text{TAG} = \{0, 1\}$, an ABO dual projective hashing degrades to a dual projective hashing (refer to Section 3.1 for details). We proceed to answer the above two problems using ABO dual projective hashing. Our applications treat u as an index and (tag, k) as an input to some hash function. It is thus more convenient to denote an ABO dual projective hashing by $\Lambda_u(tag, k)$. For clarity, we replace k with x and use $\Lambda_u(tag, x)$ instead.

ABO Lossy Trapdoor Functions. A collection of ABO lossy trapdoor functions is associated with a set, whose members are called branches. The generator of the collection takes an additional parameter $b^* \in B$, and outputs a description of a function $f(\cdot, \cdot)$ together with a trapdoor τ. The function f has the property that for any branch $b \neq b^*$ the function $f(b, \cdot)$ is injective and can be inverted using τ, while $f(b^*, \cdot)$ is lossy, which means each function statistically loses a significant amount of information about its input. Moreover, the hidden lossy branch property requires that a description of a random function f_1 generated with a parameter b_1 should be indistinguishable from a description of a random function f_2 generated with a distinct parameter b_2.

Starting from ABO dual projective hashing, we can build a collection of ABO lossy trapdoor functions as $F_{u,tag} : x \mapsto \alpha(x)||\Lambda_u(tag, x)$. The parameter u is generated by a key generation algorithm whose inputs are the projective tag tag^* together with some trapdoor information. For the injective branch $tag \neq tag^*$, invertibility guarantees that, x can be efficiently recovered from the output of the hash function. For the lossy branch tag^*, the projective property guarantees that the output is fully determined by $\alpha(x)$ (and u), and therefore preserves at most $\log|\alpha(x)|$ bits information of x. The hidden lossy branch property is implied by the hidden projective tag property of ABO dual projective hashing.

Deterministic Encryption. Deterministic public key encryption, first introduced by Bellare, Boldyreva and O'Neill [2], is proposed as an alternative in scenarios where traditional randomized encryptions exhibit inherent drawbacks, such as failure in supporting efficient search on encrypted data by simple equality test. The only known general construction of CCA secure deterministic encryption schemes was presented by Boldyreva, Fehr and O'Neill [7]. We give a new one follow their approach. The differences are, they used (ABO) lossy trapdoor functions in place of (ABO) dual projective hashing and the lossy mode acts as an

universal hash function (called universal hash mode). With a family of universal hash function $\mathcal{H}$ which is universal one-way, a dual projective hashing Λ, and an ABO dual projective hashing Λ', our construction is roughly as follows.

- The key generator chooses a random NO instance u of Λ together with a trapdoor τ, and generates a random instance u' of Λ' together with trapdoor τ' under a default projective tag. The public key is $pk = (u, u', H)$ where h is a hash function chosen at random from $\mathcal{H}$. The secret key is (τ, τ', pk).
- The encryption algorithm encrypts a message m as follows: $H(m)||\alpha(m)||\Lambda_u(m)||\alpha'(m)||\Lambda'_u(H(m), m)$. Note that Λ'_u uses $H(m)$ as tag.
- The decryption algorithm attempts to decrypt a ciphertext $c = h||c_1||c_2||c_3||c_4$ as follows: It computes m' from c_1, c_2 using the trapdoor τ. Since Λ is invertible on NO instance u, this can be done efficiently. It outputs m' if the ciphertext is well-formed, that means it can be reconstructed from m'.

We show that if both $\alpha(\cdot)$ and $\alpha'(\cdot)$ are strong average-case extractors (where the seed is provided by the public parameter) for high min-entropy sources, then we obtain a CCA secure deterministic encryption scheme for high min-entropy message distributions. With these requirements on α and α', dual and ABO dual projective hashing imply lossy and ABO lossy trapdoor functions with universal hash mode, respectively, so our construction in general follows from their framework. The additional requirements on $\alpha(\cdot)$ and $\alpha'(\cdot)$ are sometimes satisfied under the cost of efficiency (i.e., the sizes of the keys and hash value). We further present an extended general construction with improved efficiency, which eliminates the extra requirement on the (ABO) dual projective hashing, similar to existing technique [7]. In particular, we use an invertible, pairwise-independent hash functions, and then show this extension suffices to provide CCA security by applying a generalized crooked leftover hash lemma [7].

Instantiations. We present instantiations of ABO dual projective hashing from three major classes of cryptographic assumptions, consisting of Diffie-Hellman assumptions like DDH and DLIN, number-theoretic assumptions like DCR, and lattice-based assumptions like LWE. Following similar technique of [25], we rely on hashing keys to be vectors and/or matrices over $\{0,1\}^*$ (except one of the DCR-based constructions) in order to achieve the invertibility.

Our results also give a unified treatment of all known constructions of ABO lossy trapdoor functions [22,13], since they can be obtained (with slight changes) by applying our generic transformations from ABO dual projective hash to ABO lossy trapdoor functions on these instantiations. In addition, we present a new construction of ABO lossy trapdoor function based on the DLIN assumption, which can be viewed as an extension of Freeman et al.'s [13] DDH-based scheme to the DLIN setting, and therefore solves an open problem left by them[1].

We then discuss instantiations of CCA secure deterministic encryption. Due to the invertibility requirement, hashing keys k are vectors and/or matrices over

[1] As explained later, DLIN-based ABO lossy trapdoor functions can be constructed from DLIN-based lossy trapdoor functions by the parallel execution technique [22].

$\{0,1\}^*$. Regarding the general construction, in order to instantiate $\alpha(\cdot)$ and $\alpha'(\cdot)$ as average-case extractors, we resort to random linear functions where the input k are vectors and/or matrices over $\{0,1\}^*$ [21,25]. For the above reasons, our DCR-based construction and the LWE-based construction are less efficient compared with those of Boldyreva, Fehr and O'Neill [7][2]. However, our DDH-based construction achieves almost the same efficiency as theirs. In addition, we present a new construction of CCA secure deterministic encryption based on the DLIN assumption. Regarding the extended general construction, our DCR-based and LWE-based instantiations are as efficient as those in [7].

1.2 Related Work

ABO Lossy Trapdoor Functions. Peikert and Waters [22] presented general constructions of ABO lossy trapdoor functions from lossy trapdoor function using the "parallel execution" technique. As the sizes of the public key and hash value are linear to the length of the branch, this approach yields inefficient constructions. They also presented direct matrix-based constructions based on DDH and LWE assumptions. Freeman et. al. [13] then proposed new and improved instantiations of ABO lossy trapdoor functions based on DDH and DCR assumptions. Recently, Joye and Libert [18] gave a new construction of ABO lossy trapdoor function based on both the k-Quadratic Residuosity and the DDH assumptions, which achieves much shorter outputs and keys than previous DDH-based ones.

Deterministic Encryptions. Bellare et al. [2] first introduced deterministic public key encryption, formalized several notions of security, and gave a construction in the random oracle model. Later, Bellare et al. [4] and Boldyreva, Fehr and O'Neill [7] refined and extended the security notions, and presented constructions in the standard model. Especially, the latter gave general constructions of CPA/CCA secure deterministic encryption schemes, as well as efficient instantiations under number-theoretic assumptions. After that, there are several follow-up works, focusing on hard-to-invert auxiliary inputs [8,25], incrementatlity [20] (i.e., small changes in the plaintext translate into small changes in the corresponding ciphertext), multi-shot adversaries [3] (i.e., adversaries that interactively challenge the scheme with plaintext distributions depending on previous ciphertexts), bounded multi-message security [14] (i.e., the number of messages are bounded before the setup of the system but messages may be arbitrarily correlated), and impossibility for unbounded multi-message security [26].

There are two main limitations in the above work. One is *plaintext unpredictability*, which means security can be satisfied when plaintext are distributed over a large set. This limitation is inherent and essential for deterministic encryption. The other limitation is *key-independent plaintext distributions*, which means plaintext distributions are independent on the public key. It was considered to be inherent, until Raghunathan, Segev and Vadhan [23] showed that this

[2] The DDH-based construction in [7] follows the general framework, while the DCR-based and the LWE-based constructions follow the extended general framework.

limitation can be removed, with meaningful security guarantee, by relying on a randomness extraction from seed-dependent distributions. They also presented CCA secure schemes based on lossy trapdoor functions.

2 Preliminaries

Notation. If A is a deterministic algorithm, then $y := A(x)$ denotes the assignment to y of the output of A on input x. If A is a probabilistic algorithm, then $y \leftarrow_{\$} A(x)$ denotes the assignment to y of the output of A on input x with a set of uniformly random coins. We write $y := A(x; r)$ to denote the assignment to y of the output of A on input x and random coins r. A function $\mu(\cdot)$, where $\mu : \mathbb{N} \to [0, 1]$ is called *negligible* if for every positive polynomial $p(\cdot)$, for all sufficiently large $\kappa \in \mathbb{N}$, $\mu(\kappa) < 1/p(\kappa)$. We use $\mathsf{negl}(\cdot)$ to denote an unspecified negligible function.

Let $\{0,1\}^n$ be the set of n-bit strings. For a string $x \in \{0,1\}^*$, $|x|$ denotes the length of x. For a random variable X, we use notation $x \leftarrow X$ to denote that a value x is sampled according to X. For a finite set $\mathcal{X}$, we write $x \leftarrow_{\$} \mathcal{X}$ to denote the assignment to x of a uniformly randomly chosen element of $\mathcal{X}$. We use $|\mathcal{X}|$ to denote the cardinality of the set $\mathcal{X}$.

The *min-entropy* of a random variable X, denoted as $\mathrm{H}_\infty(X)$, is $\mathrm{H}_\infty(X) := -\log(\max_x \Pr[X = x])$. A *k-source* is a random variable X with $\mathrm{H}_\infty(X) \geq k$. A family of hash functions is a pair $\mathcal{H} := (\mathcal{K}, H)$ where the key generation algorithm $\mathcal{K}(1^\kappa)$ returns a key K, and the deterministic hash function H takes K and an input x to return a hash value y. Let $\ell := \ell(\kappa)$ be a polynomial-time computable function. For simplicity, $\{0,1\}^\ell$ and $\mathcal{R}$ denote the domain and image of $H(K, \cdot)$, respectively. We call $\mathcal{H}$ an *ℓ-bit hash function.* We say that an ℓ-bit hash function $\mathcal{H}$ with image $\mathcal{R}$ is *universal* if for all $x_1 \neq x_2 \in \{0,1\}^\ell$, $\Pr[H(K, x_1) = H(K, x_2) : K \leftarrow_{\$} \mathcal{K}(1^\kappa)] \leq 1/|\mathcal{R}|$. If we have an upper bound of $\epsilon < 1$ on the collision probability, we say that $\mathcal{H}$ is ϵ-almost universal. We say that $\mathcal{H}$ is *pairwise-independent* if for all $x_1 \neq x_2 \in \{0,1\}^\ell$ and $y_1, y_2 \in \mathcal{R}$, $\Pr[H(K, x_1) = y_1 \wedge H(K, x_2) = y_2 : K \leftarrow_{\$} \mathcal{K}(1^\kappa)] \leq 1/|\mathcal{R}|^2$.

We say that $\mathcal{H}$ is *universal one-way* (UOW) if for every PPT adversary $A := (A_1, A_2)$, the *UOW-advantage* $\mathsf{Adv}^{\mathrm{uow}}_{\mathcal{H},A}(\kappa) := \Pr[H(K, x_1) = H(K, x_2) : (x_1, st) \leftarrow A_1(1^\kappa), K \leftarrow_{\$} \mathcal{K}(1^\kappa), x_2 \leftarrow A_2(K, st)]$ of A is negligible in κ. We say that $\mathcal{H}$ is *collision-resistant (CR)* if for every PPT adversary A, the advantage $\mathsf{Adv}^{\mathrm{cr}}_{\mathcal{H},A}(\kappa) := \Pr[H(K, x_1) = H(K, x_2) \wedge x_1 \neq x_2 :, K \leftarrow_{\$} \mathcal{K}(1^\kappa), x_1, x_2 \leftarrow A_2(K)]$ of A is negligible in κ. UOW is implied by CR.

Definition 1 (Dual Projective Hashing). *A dual projective hashing* $\mathbf{P}$ *consists of the following polynomial-time algorithms:* $\mathsf{Setup}, \mathsf{Pub}, \mathsf{Priv}, \mathsf{Tdinv}$.

- $\mathsf{Setup}(1^\kappa)$*: takes as input a security parameter* κ *expressed in the unary representation, and generates parameterized instances of the form* $\mathsf{para} := (hp, msk, \mathcal{X}, \mathcal{Y}, \mathcal{P}, \mathcal{U} = \Pi_Y \bigcup \Pi_N, \mathcal{W}, \Gamma, \mathbf{H}, \alpha)$*, where hp contains global public parameters*[3]*, msk is a master trapdoor related to hp (e.g., the randomness*

[3] Throughout the paper, we assume that all algorithms get *hp* as an input, and sometimes omit *hp* from the input for brevity.

used to generate hp), Π_Y and Π_N are disjoint sets and correspond to YES and NO instances, respectively, $\mathbf{H} := \{\Lambda_u : \mathcal{X} \to \mathcal{Y}\}_{u \in \mathcal{U}}$ is a family of hash functions indexed by $u \in \mathcal{U}$, and $\alpha : \mathcal{X} \to \mathcal{P}$ is a projective map (that we will explain later). In addition, we require that there exists a pair of efficient sampling algorithms SampYes *and* SampNo.

  - YES instance sampling algorithm*:* $\mathsf{SampYes}(hp)$ *outputs a random pair of values (u, w) where u is uniformly distributed over Π_Y and w is the corresponding witness in $\mathcal{W}$;*
  - NO instance sampling algorithm*:* $\mathsf{SampNo}(hp)$ *outputs a random pairs of values (u, τ) where u is uniformly distributed over Π_N and τ is the corresponding trapdoor in Γ. Note that for some instantiations,* $\mathsf{SampNo}(hp)$ *requires as input the master trapdoor msk in order to compute the inversion trapdoor τ.*

– $\mathsf{Priv}(u, x)$*: is a deterministic* private evaluation algorithm. *It takes as input a public parameter $u \in \mathcal{U}$ and an input $x \in \mathcal{X}$, outputs $y \in \mathcal{Y}$.*
– $\mathsf{Pub}(u, \alpha(x), w)$*: is a deterministic* public evaluation algorithm. *It takes as input a public parameter $u \in \Pi_Y$, a projective value $\alpha(x) \in \mathcal{P}$, and a witness w for u, outputs $y \in \mathcal{Y}$.*
– $\mathsf{Tdinv}(\tau, \alpha(x), \Lambda_u(x))$*: takes as input a trapdoor information $\tau \in \Gamma$, a projective value $\alpha(x) \in \mathcal{P}$ for any $x \in \mathcal{X}$, and a hash value $\Lambda_u(x) \in \mathcal{Y}$, outputs $x' \in \mathcal{X}$.*

Correctness. We require that for all $\kappa \in \mathbb{N}$, all para generated by $\mathsf{Setup}(1^\kappa)$, all $u \in \Pi_Y \bigcup \Pi_N$ and all $x \in \mathcal{X}$, $\mathsf{Priv}(u, x) = \Lambda_u(x)$.

Projectiveness. $\mathbf{P}$ is *almost projective* if for all $\kappa \in \mathbb{N}$, all para generated by $\mathsf{Setup}(1^\kappa)$, all $x \in \mathcal{X}$, $\Pr[\mathsf{Pub}(u, \alpha(x), w) = \Lambda_u(x) : (u, w) \leftarrow_\$ \mathsf{SampYes}(hp)] \geq 1 - \mathsf{negl}(\kappa)$. If this holds with probability 1, we say that $\mathbf{P}$ is *perfectly projective.*

Invertibility. $\mathbf{P}$ is *almost invertible* if for all $\kappa \in \mathbb{N}$, all para generated by $\mathsf{Setup}(1^\kappa)$, all $x \in \mathcal{X}$, $\Pr[\mathsf{Tdinv}(\tau, \alpha(x), \Lambda_u(x)) = x : (u, \tau) \leftarrow_\$ \mathsf{SampNo}(hp)] \geq 1 - \mathsf{negl}(\kappa)$. If this holds with probability 1, we say that $\mathbf{P}$ is *perfectly invertible.*
Subset Membership Assumption. This assumption states that the uniform distributions over Π_Y and Π_N are computationally indistinguishable, even given hp. This is formally captured by the advantage function $\mathsf{Adv}^{\mathrm{sm}}_{\mathrm{DPH},A}(\kappa)$:

$$\mathsf{Adv}^{\mathrm{sm}}_{\mathrm{DPH},A}(\kappa) := \Pr[A(hp, u) = 1 : u \leftarrow_\$ \Pi_Y] - \Pr[A(hp, u) = 1 : u \leftarrow_\$ \Pi_N]$$

where hp is generated by $\mathsf{Setup}(1^\kappa)$. The subset membership assumption states that for all PPT adversary A, $\mathsf{Adv}^{\mathrm{sm}}_{\mathrm{DPH},A}(\kappa)$ is a negligible function in κ.

3 ABO Dual Projective Hashing

Definition 2 (ABO Dual Projective Hashing). *An all-but-one dual projective hashing* $\mathbf{P}$ *consists of the following polynomial-time algorithms:* Setup, Keygen, Pub, Priv, Tdinv.

- $\mathsf{Setup}(1^\kappa)$*: takes as input a security parameter* κ *expressed in the unary representation, and generates parameterized instances of the form* $\mathsf{para} := (hp, msk, \text{TAG}, \mathcal{X}, \mathcal{Y}, \mathcal{P}, \mathcal{U}, \mathcal{W}, \Gamma, \mathbf{H}, \alpha)$*, where* hp *contains global public parameters,* msk *is a master trapdoor related to* hp *(e.g., the randomness used to generate* hp*),* $\mathbf{H} := \{\Lambda_u : \text{TAG} \times \mathcal{X} \to \mathcal{Y}\}_{u \in \mathcal{U}}$ *is a family of hash functions indexed by* $u \in \mathcal{U}$*, and* $\alpha : \mathcal{X} \to \mathcal{P}$ *is a projective map.*
- $\mathsf{Keygen}(msk, tag^*)$*: takes as input a master trapdoor* msk *and a tag* $tag^* \in$ TAG*, and outputs* (u, w, τ) *consisting of a public parameter* $u \in \mathcal{U}$*, a witness* $w \in \mathcal{W}$*, and an inversion trapdoor* $\tau \in \Gamma$*.* If no tag input is specified, it is assumed to be a fixed "default" tag.
- $\mathsf{Priv}(u, tag, x)$*: is the deterministic* private evaluation algorithm*. It takes as input a public parameter* $u \in \mathcal{U}$*, a tag* $tag \in$ TAG *and an input* $x \in \mathcal{X}$*, and outputs* $y \in \mathcal{Y}$*.*
- $\mathsf{Pub}(u, tag, \alpha(x), w)$*: is the deterministic* public evaluation algorithm*. It takes as input a public parameter* $u \in \mathcal{U}$*, a tag* $tag \in$ TAG*, a projective value* $\alpha(x) \in \mathcal{P}$ *and a witness* w*, and outputs* $y \in \mathcal{Y}$ *if* $tag = tag^*$*.*
- $\mathsf{Tdinv}(\tau, tag, \alpha(x), \Lambda_u(tag, x))$*: takes as input a trapdoor information* $\tau \in \Gamma$*, a tag* $tag \in$ TAG*, a projective value* $\alpha(x) \in \mathcal{P}$ *for any* $x \in \mathcal{X}$*, and a hash value* $\Lambda_u(tag, x)$ *for some tag* $tag \in$ TAG*, and outputs* $x' \in \mathcal{X}$ *if* $tag \neq tag^*$*.*

Correctness. We require that for all $\kappa \in \mathbb{N}$, all para generated by $\mathsf{Setup}(1^\kappa)$, all $tag^* \in$ TAG, all (u, w, τ) generated by $\mathsf{Keygen}(msk, tag^*)$, and all $x \in \mathcal{X}$, $\mathsf{Priv}(u, tag, x) = \Lambda_u(tag, x)$.

Projectiveness. We say $\mathbf{P}$ is *almost projective* if for all $\kappa \in \mathbb{N}$, all para generated by $\mathsf{Setup}(1^\kappa)$, all $tag^* \in$ TAG, all $x \in \mathcal{X}$, $\Pr[\mathsf{Pub}(u, tag^*, \alpha(x), w) = \Lambda_u(tag^*, x) : (u, w, \tau) \leftarrow_\$ \mathsf{Keygen}(msk, tag^*)] \geq 1 - \mathsf{negl}(\kappa)$. If the projective property holds with probability 1 then we say that $\mathbf{P}$ is *perfectly projective*.

Invertibility. We say $\mathbf{P}$ is *almost invertible* if for all $\kappa \in \mathbb{N}$, all $\mathsf{para} \leftarrow \mathsf{Setup}(1^\kappa)$, all $tag^*, tag \in$ TAG where $tag^* \neq tag$, all $x \in \mathcal{X}$, $\Pr[\mathsf{Tdinv}(\tau, tag, \alpha(x), \Lambda_u(tag, x)) = x : (u, w, \tau) \leftarrow_\$ \mathsf{Keygen}(msk, tag^*)] \geq 1 - \mathsf{negl}(\kappa)$. If the invertibility holds with probability 1 then we say that $\mathbf{P}$ is *perfectly invertible*.

Hidden Projective Tag. For every para generated by $\mathsf{Setup}(1^\kappa)$ and for any PPT algorithm $A := (A_1, A_2)$, the advantage $\mathsf{Adv}^{\mathrm{hpt}}_{\mathbf{P},A}(\kappa)$ of A is negligible in the security parameter κ:

$$\mathsf{Adv}^{\mathrm{hpt}}_{\mathbf{P},A}(\kappa) := 2 \Pr\left[b = b' : \begin{array}{l}((tag_0, tag_1), st) \leftarrow A_1(hp), b \leftarrow_\$ \{0,1\} \\ (u, w, \tau) \leftarrow_\$ \mathsf{Keygen}(msk, tag_b), b' \leftarrow A_2(hp, u, st)\end{array}\right] - 1.$$

Dual projective hashing (DPH) and ABO dual projective hashing are equivalent for appropriate choices of parameters. We show their relationship in Section 3.1.

3.1 Relationship between DPH and ABO DPH

From ABO DPH to DPH. Starting from an ABO dual projective hashing $\mathbf{P} := (\mathsf{Setup}, \mathsf{Keygen}, \mathsf{Pub}, \mathsf{Priv}, \mathsf{Tdinv})$ with tag set TAG $= \{0, 1\}$, we may derive a dual projective hashing as follows.

- $\mathsf{Setup}'(1^\kappa)$: runs $(hp, msk, \{0,1\}, \mathcal{X}, \mathcal{Y}, \mathcal{P}, \mathcal{U}, \mathcal{W}, \Gamma, \mathbf{H}, \alpha) \leftarrow_\$ \mathsf{Setup}(1^\kappa)$, then run $(u_0, w_0, \tau_0) \leftarrow_\$ \mathsf{Keygen}(msk, 0)$, and $(u_1, w_1, \tau_1) \leftarrow_\$ \mathsf{Keygen}(msk, 1)$. Denote by Π_Y and Π_N the set of possible value of u_0 and u_1, respectively. The family of functions $\mathbf{H}' := \{\Lambda'_u : \mathcal{X} \to \mathcal{Y}\}_{u \in \mathcal{U}'}$ is defined as $\Lambda'_u(x) := \Lambda_u(0, x)$. Return $(hp, msk, \mathcal{X}, \mathcal{Y}, \mathcal{P}, \mathcal{W}, \mathcal{U}' := \Pi_Y \bigcup \Pi_N, \Gamma, \mathbf{H}', \alpha)$.
  - $\mathsf{SampYes}(hp)$: runs $(u, w, \tau) \leftarrow_\$ \mathsf{Keygen}(msk, 0)$ and outputs (u, w).
  - $\mathsf{SampNo}(hp)$: runs $(u, w, \tau) \leftarrow_\$ \mathsf{Keygen}(msk, 1)$ and outputs (u, τ).
- $\mathsf{Priv}'(u, x)$: outputs $\mathsf{Priv}(u, 0, x)$.
- $\mathsf{Pub}'(u, \alpha(x), w)$: outputs $\mathsf{Pub}(u, 0, \alpha(x), w)$.
- $\mathsf{Tdinv}'(\tau, \alpha(x), y)$: outputs $x \leftarrow \mathsf{Tdinv}(\tau, 0, \alpha(x), y)$.

From DPH to ABO DPH. We give a general construction of ABO dual projective hashing from a dual projective hashing by "parallel execution" which has been used in previous works [12,24,22]. Starting from a dual projective hashing $\mathbf{P} := (\mathsf{Setup}, \mathsf{Pub}, \mathsf{Priv}, \mathsf{Tdinv})$, we can derive an ABO dual projective hashing for tag set $\{0,1\}^\ell$ as follows.

- $\mathsf{Setup}'(1^\kappa)$: runs $(hp, msk, \mathcal{X}, \mathcal{Y}, \mathcal{P}, \mathcal{U} = \Pi_Y \bigcup \Pi_N, \mathcal{W}, \Gamma, \mathbf{H}, \alpha) \leftarrow_\$ \mathsf{Setup}(1^\kappa)$. Sets $\mathrm{TAG} := \{0,1\}^\ell$. Sets $\mathcal{Y}' := \mathcal{Y}^\ell, \mathcal{U}' := \mathcal{U}^{2\ell}, \mathcal{W}' := \mathcal{W}^\ell$, $\Gamma' := \Gamma^\ell$. The family of functions $\mathbf{H}' := \{\Lambda'_{u'} : \mathrm{TAG} \times \mathcal{X} \to \mathcal{Y}'\}_{u' \in \mathcal{U}'}$ is defined as $\Lambda'_{u'}(tag, x) := (\Lambda_{u_{i,tag_i}}(x))_{i \in [\ell]}$ where u' equals $(u_{i,0}, u_{i,1})_{i \in [\ell]}$. Returns $(hp, msk, \mathrm{TAG}, \mathcal{X}, \mathcal{Y}', \mathcal{P}, \mathcal{U}', \mathcal{W}', \Gamma', \mathbf{H}', \alpha)$.
- $\mathsf{Keygen}'(msk, tag^*)$: for $i = 1$ to ℓ, runs $(u_{i,tag_i^*}, w_i) \leftarrow_\$ \mathsf{SampYes}(hp)$ and $(u_{i,1-tag_i^*}, \tau_i) \leftarrow_\$ \mathsf{SampNo}(hp)$. Sets $u' := (u_{i,0}, u_{i,1})_{i \in [\ell]}$, $w' := (w_i)_{i \in [\ell]}$, and $\tau' := (\tau_i)_{i \in [\ell]}$. Outputs (u', w', τ').
- $\mathsf{Priv}'(u', tag, x)$: parses u' as $(u_{i,0}, u_{i,1})_{i \in [\ell]}$, and outputs $(\mathsf{Priv}(u_{i,tag_i}, x))_{i \in [\ell]}$.
- $\mathsf{Pub}'(u', tag, \alpha(x), w')$: if $tag \neq tag^*$, outputs $\perp$. Otherwise parses u' as $(u_{i,0}, u_{i,1})_{i \in [\ell]}$ and w' as $(w_i)_{i \in [\ell]}$, and outputs $(\mathsf{Pub}(u_{i,tag_i}, \alpha(x), w_i))_{i \in [\ell]}$.
- $\mathsf{Tdinv}'(\tau', tag, \alpha(x), (y_1, \ldots, y_\ell))$: computes $x_i \leftarrow \mathsf{Tdinv}(\tau_i, \alpha(x), y_i)$ for all i such that $tag_i \neq tag_i^*$. Denote the common value by x if all these values agree and if not outputs $\perp$. Checks $y_i = \mathsf{Priv}(u_{i,tag_i}, x)$ for all i such that $tag_i = tag_i^*$. If all the checks pass, then outputs x; otherwise outputs $\perp$.

4 All-but-One Lossy Trapdoor Functions from ABO DPH

We construct a family of ABO lossy trapdoor functions in Fig. 1.

Theorem 1. *Suppose that* $\mathbf{P} := (\mathsf{Setup}, \mathsf{Keygen}, \mathsf{Pub}, \mathsf{Priv}, \mathsf{Tdinv})$ *is an ABO dual projective hashing, then the construction in Fig. 1 yields a collection of* $(m, m - \log|\mathrm{Img}\alpha|)$*-ABO lossy trapdoor functions, where* $m := \log|\mathcal{X}|$.

Proof. The correctness for injective functions follows from the invertibility property. The lossiness for the lossy branch follows from the projective property. Recall that if $tag = tag^*$, then for all $x \in \mathcal{X}$, $\Lambda_u(tag, x)$ is determined by $\alpha(x)$ and u. This means that the size of image set $\mathrm{Img} f_{u,tag}$ is at most $|\mathrm{Img}\alpha|$. Thus, the function is $(m, m - \log|\mathrm{Img}\alpha|)$-lossy. The hidden lossy branch property directly follows from the hidden projective tag property of $\mathbf{P}$.

All-but-One Lossy Trapdoor Function

1. *Sampling a branch*: $B(1^\kappa)$ outputs a value $tag^* \in \text{TAG}$.
2. *Sampling a function*: $S_{\text{abo}}(1^\kappa, tag^*)$ first runs $(u, w, \tau) \leftarrow_\$ \mathsf{Keygen}(msk, tag^*)$, and outputs $(hp||u, \tau)$.
3. *Evaluation*: $F_{\text{abo}}(hp||u, tag, x)$ returns $\alpha(x)||\Lambda_u(tag, x)$. Note $\Lambda_u(tag, x)$ can be computed using $\mathsf{Priv}(u, tag, x)$.
4. *Inversion of injective functions*: Returns $\mathsf{Tdinv}(\tau, tag, \alpha(x), \Lambda_u(tag, x))$ if $tag \neq tag^*$.

Note: $(hp, msk, \text{TAG}, \mathcal{X}, \mathcal{Y}, \mathcal{P}, \mathcal{U}, \mathcal{W}, \Gamma, \mathbf{H}, \alpha) \leftarrow_\$ \mathsf{Setup}(1^\kappa)$.

Fig. 1. ABO lossy trapdoor function from ABO dual projective hashing

5 Deterministic Encryption from ABO DPH

5.1 Security Definition

Under page limit, we omit the definition of extractors and the left-over hash lemma. Next we give the definition of deterministic encryption.

Definition 3 (Deterministic Encryption). *A deterministic encryption scheme Π is specified by three polynomial-time algorithms, $\mathsf{Gen}, \mathsf{Enc}$ and Dec.*

- $\mathsf{Gen}(1^\kappa)$*: on input a security parameter κ expressed in the unary representation, the key generation algorithm outputs a public key pk and a secret key sk. The pk includes a description of finite message space $\mathcal{M}$ and a finite ciphertext space $\mathcal{C}$.*
- $\mathsf{Enc}(pk, m)$*: on input pk and a message $m \in \mathcal{M}$, the* deterministic *encryption algorithm outputs a ciphertext $c \in \mathcal{C}$.*
- $\mathsf{Dec}(sk, c)$*: on input a secret key sk and a ciphertext c, the decryption algorithm outputs a message $m \in \mathcal{M} \cup \perp$.*

Correctness. For all $\kappa \in \mathbb{N}$, all message $m \in \mathcal{M}$, it holds that

$$\Pr\left[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) \neq m : (pk, sk) \leftarrow_\$ \mathsf{Gen}(1^\kappa)\right] \leq \mathsf{negl}(\kappa).$$

Security under chosen-ciphertext attack. We follow the indistinguishability-based security definition of deterministic encryption [7,4]. For simplicity, we only consider security while encrypting a single message, although our proof extends to multiple messages for block-sources. We can also rely on the existing result that for block-sources, single message security equals to multi-message security [7].

Definition 4 (PRIV-CCA). *A deterministic encryption $\Pi := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is* PRIV-CCA*-secure for k-source if for any k-source $\mathsf{M}_0, \mathsf{M}_1$, the advantage $\mathsf{Adv}^{priv\text{-}cca}_{\Pi,A,\mathsf{M}_0,\mathsf{M}_1}(\kappa) := 2\Pr[\mathsf{Exp}^{priv\text{-}cca}_{\Pi,A,\mathsf{M}_0,\mathsf{M}_1}(\kappa) = 1] - 1$ of any* PPT *adversary A is negligible in κ. The experiment $\mathsf{Exp}^{priv\text{-}cca}_{\Pi,A,\mathsf{M}_0,\mathsf{M}_1}(\kappa)$ is defined by: 1) $b \leftarrow_\$ \{0,1\}$; 2) $m_b \leftarrow \mathsf{M}_b, (pk, sk) \leftarrow_\$ \mathsf{Gen}(1^\kappa)$; 3) $c := \mathsf{Enc}(pk, m_b)$; 4) $b' \leftarrow A^{\mathsf{Dec}_{\neq c}(sk,\cdot)}(pk, c)$ where the oracle $\mathsf{Dec}_{\neq c}(sk, \cdot)$ decrypts any ciphertext except c; 5) Return $b = b'$.*

5.2 Our Construction

Let $\mathbf{P} := (\mathsf{Setup}, \mathsf{Pub}, \mathsf{Priv}, \mathsf{Tdinv})$, $\mathbf{P}' := (\mathsf{Setup}', \mathsf{Keygen}', \mathsf{Pub}', \mathsf{Priv}', \mathsf{Tdinv}')$ be a dual projective hashing and an ABO dual projective hashing respectively. Let $\mathcal{H} := (\mathcal{K}, H)$ be an ℓ-bit universal and universal one-way hash function with image $\mathcal{R}$. For consistency, $\mathcal{R}$ does not include the default projective tag tag^* of $\mathbf{P}'$. The deterministic encryption Π is shown in Fig. 2. The message space $\mathcal{M}$ is a subset of both $\mathcal{X}$ and $\mathcal{X}'$, and the image $\mathcal{R}$ of the hash function $\mathcal{H}$ is a subset of the set $\text{TAG}'\backslash\{tag^*\}$.

Key Generation: $\mathsf{Gen}(1^\kappa)$ computes as follows.

1. Run $(hp, msk, \mathcal{X}, \mathcal{Y}, \mathcal{P}, \mathcal{U}, \mathcal{W}, \Gamma, \mathbf{H}, \alpha) \leftarrow_\$ \mathsf{Setup}(1^\kappa)$.
2. Run $(hp', msk', \text{TAG}', \mathcal{X}', \mathcal{Y}', \mathcal{P}', \mathcal{U}', \mathcal{W}', \Gamma', \mathbf{H}', \alpha') \leftarrow_\$ \mathsf{Setup}'(1^\kappa)$.
3. Run $(u, \tau) \leftarrow_\$ \mathsf{SampNo}(hp)$, $(u', w', \tau') \leftarrow_\$ \mathsf{Keygen}'(msk', tag^*)$, and $K \leftarrow_\$ \mathcal{K}(1^\kappa)$.
4. Output $pk := hp||u||hp'||u'||K$ and $sk := \tau||\tau'||w'||pk$.

Encryption: $\mathsf{Enc}(pk, m)$ takes input $pk = hp||u||hp'||u'||K$ and message m, and computes as follows.

1. $h := H(K, m)$.
2. $c_1 := \alpha(hp, m)$ and $c_2 := \Lambda_u(m)$. Note that c_2 can be computed using $\mathsf{Priv}(u, m)$.
3. $c_3 := \alpha'(hp', m)$ and $c_4 := \Lambda'_{u'}(h, m)$. Note that c_4 can be computed using $\mathsf{Priv}'(u', h, m)$.
4. Output $h||c_1||c_2||c_3||c_4$.

Decryption: $\mathsf{Dec}(sk, c)$ computes as follows.

1. Parse sk as $\tau||\tau'||w'||pk$ and c as $h||c_1||c_2||c_3||c_4$.
2. $m' \leftarrow \mathsf{Tdinv}(\tau, c_1, c_2)$.
3. $c' := \mathsf{Enc}(pk, m')$.
4. If $c = c'$ then return m'; otherwise return $\perp$.

Fig. 2. Deterministic encryption scheme from (ABO) dual projective hashing

Theorem 2. *Suppose that* $(x, hp) \longmapsto \alpha(hp, x)$ *is an average-case* (k_1, ϵ_1)*-extractor,* $(x, hp') \longmapsto \alpha'(hp', x)$ *is an average-case* (k_2, ϵ_2)*-extractor, the subset membership assumption for* $\mathbf{P}$ *holds, and* $\mathcal{H} := (\mathcal{K}, H)$ *is* ℓ*-bit universal hash function that is also universal one-way. For any adversary* A*, any* k*-sources* $\mathsf{M}_0, \mathsf{M}_1$ *such that* $k \geq \max\{k_1 + \log|\mathcal{R}|, k_2 + \log|\mathcal{R}| + \log|\mathcal{P}|, \log|\mathcal{R}| + 2\log(1/\epsilon_3)\}$*, there exist adversaries* B_{hpt}, B_{uow}, B_{sm} *such that:*

$$\mathsf{Adv}^{priv\text{-}cca}_{\Pi, A, \mathsf{M}_0, \mathsf{M}_1}(\kappa) \leq 2\big(\mathsf{Adv}^{hpt}_{\mathbf{P}, B_{hpt}}(\kappa) + \mathsf{Adv}^{uow}_{\mathcal{H}, B_{uow}}(\kappa) + \mathsf{Adv}^{sm}_{\text{DPH}, B_{sm}}(\kappa) + \epsilon_1 + \epsilon_2 + \epsilon_3\big).$$

Furthermore, the running-time of B_{hpt}, B_{uow}, B_{sm} *are roughly that of* A*.*

5.3 Extended General Construction

Our security proofs explored the fact that the projective map α acts as an average-case extractor. In specific instantiations, we actually design α as a universal hash function and then apply the generalized leftover hash lemma (LHL)

to conclude it is an average-case extractor. This sometimes results in inefficient constructions. Using similar technique of [7], we present an extension of our generic construction, where the extra universality requirement on α is eliminated. We use an invertible, pairwise-independent hash functions, and then showed this extension suffices to provide CCA security by applying a generalized crooked LHL [7].

We say a family of pairwise-independent hash functions $\mathcal{H}_{pi} := (\mathcal{K}_{pi}, H_{pi})$ is *invertible* if there is a PPT algorithm I such that for all K_{pi} output by $\mathcal{K}_{pi}$ and all $m \in \{0,1\}^{\ell}$, $I(K_{pi}, H_{pi}(K_{pi}, m))$ outputs m. Let $\mathbf{P} := (\mathsf{Setup}, \mathsf{Pub}, \mathsf{Priv}, \mathsf{Tdinv})$ be a dual projective hashing. Let $\mathbf{P}' := (\mathsf{Setup}', \mathsf{Keygen}', \mathsf{Pub}', \mathsf{Priv}', \mathsf{Tdinv}')$ be an ABO dual projective hashing. Let $\mathcal{H}_{pi} := (\mathcal{K}_{pi}, H_{pi})$ be a family of ℓ-bit invertible pairwise-independent permutations on $\{0,1\}^{\ell}$. For consistency, $\mathcal{H}_{pi}$ does not map to a default projective tag tag^* of $\mathbf{P}'$. Let $\mathcal{H}_{\text{uow}} := (\mathcal{K}_{\text{uow}}, H)$ be a family of universal one-way hash function with image $\mathcal{R}_{\text{uow}}$. The extended generation construction of deterministic encryption scheme $\Pi := (\mathsf{Enc}^+, \mathsf{Gen}^+, \mathsf{Dec}^+)$ is shown in Fig. 3. The message space $\mathcal{M}$ is $\{0,1\}^{\ell}$. The image of $\mathcal{H}_{pi}$ is a subset of $\mathcal{X}, \mathcal{X}'$, and the domain of $\mathcal{H}_{\text{uow}}$.

Key Generation: $\mathsf{Gen}^+(1^{\kappa})$ computes as follow.

1. Run $(hp, msk, \mathcal{X}, \mathcal{Y}, \mathcal{P}, \mathcal{U}, \mathcal{W}, \Gamma, \mathbf{H}, \alpha) \leftarrow_{\$} \mathsf{Setup}(1^{\kappa})$.
2. Run $(hp', msk', \text{TAG}', \mathcal{X}', \mathcal{Y}', \mathcal{P}', \mathcal{U}', \mathcal{W}', \Gamma', \mathbf{H}', \alpha') \leftarrow_{\$} \mathsf{Setup}'(1^{\kappa})$.
3. Run $(u, \tau) \leftarrow_{\$} \mathsf{SampNo}(hp), (u', w', \tau') \leftarrow_{\$} \mathsf{Keygen}'(msk', tag^*), K_{\text{uow}} \leftarrow_{\$} \mathcal{K}_{\text{uow}}(1^{\kappa})$.
4. For $i = 1$ to 3 do $K_{pi,i} \leftarrow_{\$} \mathcal{K}_{pi}(1^{\kappa})$.
5. Output $pk := hp||u||hp'||u'||K_{\text{uow}}||K_{pi,1}||K_{pi,2}||K_{pi,3}$ and $sk := \tau||\tau'||w'||pk$.

Encryption: $\mathsf{Enc}^+(pk, m)$ takes input $pk = hp||u||hp'||u'||K_{\text{uow}}||K_{pi,1}||K_{pi,2}||K_{pi,3}$ and message m, and computes as follows.

1. For $i = 1$ to 3 do $h_i := H_{pi}(K_{pi,i}, m)$.
2. $h := H(K_{\text{uow}}, h_1)$.
3. $c_1 := \alpha(hp, h_2)$ and $c_2 := \Lambda_u(h_2)$. Note that c_2 can be computed using $\mathsf{Priv}(u, h_2)$.
4. $c_3 := \alpha'(hp', h_3)$ and $c_4 := \Lambda'_{u'}(h, h_3)$. Note that c_4 can be computed using $\mathsf{Priv}'(u', h, h_3)$.
5. Output $h||c_1||c_2||c_3||c_4$.

Decryption: $\mathsf{Dec}^+(sk, c)$ computes as follows.

1. Parse sk as $\tau||\tau'||w'||pk$ and c as $h||c_1||c_2||c_3||c_4$.
2. $h'_2 \leftarrow \mathsf{Tdinv}(\tau, c_1, c_2)$.
3. $m' \leftarrow I(K_{pi,2}, h'_2)$.
4. $c' := \mathsf{Enc}^+(pk, m')$.
5. If $c = c'$ then return m'; otherwise return $\perp$.

Fig. 3. Deterministic encryption scheme from (ABO) dual projective hashing

Using the generalized crooked LHL [7], we are able to show the following.

Theorem 3. *Let $\Pi := (\mathsf{Enc}^+, \mathsf{Gen}^+, \mathsf{Dec}^+)$ be as defined in Fig. 3. For any adversary A, any k-sources $\mathsf{M}_0, \mathsf{M}_1$ such that $k \geq \log|\mathcal{R}_{uow}| + \log|\mathcal{P}| + \log|\mathcal{P}'| + 2\log(1/\epsilon) - 2$, there exist adversaries B_{hpt}, B_{uow}, B_{sm} such that $\mathsf{Adv}^{priv\text{-}cca}_{\Pi,A,\mathsf{M}_0,\mathsf{M}_1}(\kappa) \leq$*

$2\big(\mathsf{Adv}^{hpt}_{\mathbf{P},B_{hpt}}(\kappa)+\mathsf{Adv}^{uow}_{\mathcal{H}_{uow},B_{uow}}(\kappa)+\mathsf{Adv}^{sm}_{\mathrm{DPH},B_{sm}}(\kappa)+3\epsilon\big)$. *Furthermore, the running-time of* B_{hpt}, B_{uow}, B_{sm} *are roughly that of* A.

6 Instantiations

6.1 Instantiations from DDH and DLIN

Let G be a finite cyclic group of prime order q specified by a randomly chosen generator g. The d-LIN assumption asserts that $g_{d+1}^{r_1+\cdots+r_d}$ is pseudorandom given $g_1,\ldots,g_{d+1},g_1^{r_1},\ldots,g_d^{r_d}$ where $g_1,\ldots,g_{d+1}\leftarrow_{\$}G; r_1,\ldots,r_d\leftarrow_{\$}\mathbb{Z}_q$.

Here we present the DLIN-based ABO dual projective hashing. When instantiated with our generic transformations, this yields a new DLIN-based $(m, m - d\log q)$-ABO lossy trapdoor functions. It also yields a similar DDH-based ABO lossy trapdoor functions as given in [13]. As the projective map α is a universal hash function, it is also an average-case extractor by applying the generalized LHL [7]. Combining the DLIN-based dual projective hashing [25] and discrete-logarithm based hash function [7] which is universal and collision-resistant, we get a new DLIN-based PRIV-CCA secure deterministic encryption scheme.

- $\mathsf{Setup}(1^\kappa)$: choose G, q, g as above and $\mathbf{P}\leftarrow_{\$}\mathbb{Z}_q^{d\times m}$. Set $hp := (G, q, g^{\mathbf{P}})$, $msk := \mathbf{P}$, $\mathcal{X} := \{0,1\}^m$, $\mathcal{Y} := G^m$, $\mathcal{P} := G^d$, $\mathcal{U} := G^{m\times m}$, $\mathcal{W} := \mathbb{Z}_q^m$, $\text{TAG} := \mathbb{Z}_q$. The map α is defined by $\alpha(g^{\mathbf{P}},\mathbf{x}) := g^{\mathbf{Px}}$ with $\mathbf{x}\in\{0,1\}^m$.
- $\mathsf{Keygen}(msk, b^*)$: choose $\mathbf{W}\leftarrow_{\$}\mathbb{Z}_q^{m\times d}$, and compute $\mathbf{U} := g^{\mathbf{WP}-b^*\mathbf{I}_m}$. The witness is $\mathbf{W}$. The inversion trapdoor is $(\mathbf{P},\mathbf{W},b^*)$. Output $(\mathbf{U},\mathbf{W},(\mathbf{P},\mathbf{W},b^*))$.
- $\mathsf{Priv}(\mathbf{U}, b, \mathbf{x})$: Compute $\Lambda_{\mathbf{U}}(b,\mathbf{x}) := \mathbf{U}^{\mathbf{x}} * g^{b\mathbf{x}}$, where $*$ indicates the component-wise product of elements of G^m.
- $\mathsf{Pub}(\mathbf{U}, b^*, g^{\mathbf{Px}}, \mathbf{W}))$: Compute $g^{\mathbf{W}(\mathbf{Px})}$.
- $\mathsf{Tdinv}((\mathbf{P},\mathbf{W},b^*), b, g^{\mathbf{Px}}, \Lambda_{\mathbf{U}}(b,\mathbf{x}))$: first compute $\mathbf{A} := \mathbf{WP} + (b - b^*)\mathbf{I}_m$. The trapdoor is $\mathbf{A}^{-1}$. Note that $\Lambda_{\mathbf{U}}(b,\mathbf{x}) = \mathbf{U}^{\mathbf{x}} * g^{b\mathbf{x}} = g^{\mathbf{Ax}}$. Given $\mathbf{A}^{-1}$, $\Lambda^*_{\mathbf{U}}(b,\mathbf{x})$, we can compute $g^{\mathbf{x}}$ and thus $\mathbf{x}$.

Projectiveness: When $\mathbf{U} = g^{\mathbf{WP}-b^*\mathbf{I}_m}$ and $b = b^*$, let $(\mathbf{U}^{\mathbf{x}})_i := \Sigma_{j=1}^m \mathbf{U}_{ij}^{x_j}$, we have $\mathsf{Priv}(\mathbf{U}, b^*, \mathbf{x}) = \mathbf{U}^{\mathbf{x}} * g^{b^*\mathbf{x}} = g^{(\mathbf{WP}-b^*\mathbf{I}_m)\mathbf{x}} * g^{b^*\mathbf{x}} = g^{\mathbf{W}(\mathbf{Px})} = \mathsf{Pub}(\mathbf{U}, b^*, g^{\mathbf{Px}}, \mathbf{W})$.

6.2 Instantiations from DCR

Fix a Blum integer $N := PQ$ for safe primes $P, Q \equiv 3 \pmod 4$ (such that $P := 2p+1$ and $Q := 2q+1$ for sufficiently large primes p, q), where N is a κ-bit string. Let $s \in \mathbb{Z}^+$ be an integer. The multiplicative group $\mathbb{Z}^*_{N^{s+1}}$ is isomorphic to $\mathbb{Z}_{N^s} \times \mathbb{Z}^*_N$. The decisional composite residuosity (DCR) assumption states that any PPT algorithm that receives an input a κ-bit N (generated as above) cannot distinguish a random element in $\mathbb{Z}^*_{N^{s+1}}$ from a random N^s-th power in $\mathbb{Z}^*_{N^{s+1}}$ with non-negligible probability of κ.

First Construction. We present the DCR-based ABO dual projective hashing, extended to the Damgård-Jurik scheme [11]. When instantiated with our generic transformation, this yields the DCR-based $(s \log N, s \log N - \log |\phi(N)|)$-ABO lossy trapdoor functions given in [13] (with slight modifications). As the projective map α is not an average-case extractor, we have to rely on the extended general framework in Section 5.3 to construct PRIV-CCA secure deterministic encryptions. By combining the dual projective hash from DCR [25] and the collision resistant hash function from DCR [7], we get a PRIV-CCA secure deterministic encryption which is as efficient as the DCR-based construction in [7].

- $\mathsf{Setup}(1^\kappa)$: choose a Blum integer $N := PQ$ as above. Pick $g \leftarrow_\$ \mathbb{Z}^*_{N^{s+1}}$. Set $hp := (N, g^{N^s})$, $msk := (g, P, Q)$, $\mathcal{X} := \mathbb{Z}_{N^s}$, $\mathcal{Y} := \mathbb{Z}^*_{N^{s+1}}$, $\mathcal{P} \subseteq \mathbb{Z}^*_{N^{s+1}}$ ($\mathcal{P}$ is isomorphic to $\mathbb{Z}^*_N$), $\mathcal{U} := \mathbb{Z}^*_{N^{s+1}}$, $\mathcal{W} := \mathbb{Z}_{N^s}$, $\text{TAG} := \{0, \ldots, 2^{\kappa/2-1}\}$. The projective map α is defined by $\alpha(g^{N^s}, x) := g^{N^s x}$ where $x \in \mathbb{Z}_{N^s}$.
- $\mathsf{Keygen}(msk, b^*)$: choose $w \leftarrow_\$ \mathbb{Z}_{N^s}$, compute public parameter $u := (1 + N)^{-b^*} \cdot g^{N^s w}$. The witness is w. The inversion trapdoor is (P, Q, b^*).
- $\mathsf{Priv}(u, b, x)$: compute $\Lambda_u(b, x) := ((1 + N)^b \cdot u)^x$.
- $\mathsf{Pub}(u, b^*, g^{N^s x}, w)$: compute $(g^{N^s x})^w$.
- $\mathsf{Tdinv}((P, Q, b^*), b, g^{N^s x}, \Lambda_u(b, x))$: observe that $\Lambda_u(b, x) = ((1+N)^b \cdot u)^x = ((1 + N)^{b-b^*} \cdot g^{N^s w})^x$. Given the inversion trapdoor (i.e., the factorization of N and the projective tag b^*), we can efficiently compute $(b - b^*)x$. In addition, the restriction $b, b^* \in \{0, \ldots, 2^{\kappa/2} - 1\}$ implies that $(b - b^*)$ is smaller than both P and Q and is therefore relatively prime to N. Thus, we can recover x by computing $(b - b^*)x \cdot (b - b^*)^{-1} \mod N^s$.

Projectiveness: When $u = (1 + N)^{-b^*} \cdot g^{N^s w}$ and $b = b^*$, we have

$$\begin{aligned}\mathsf{Priv}(u, b^*, x) = ((1 + N)^{b^*} \cdot u)^x &= ((1 + N)^{b^*} \cdot (1 + N)^{-b^*} \cdot g^{N^s w})^x \\ &= g^{N^s wx} = (g^{N^s x})^w = \mathsf{Pub}(u, b^*, g^{N^s x}, w).\end{aligned}$$

The uniform distributions over $\{(1 + N)^{-b} \cdot g^{N^s w} : w \in \mathbb{Z}_{N^s}\}$ and $\{g^{N^s w} : w \in \mathbb{Z}_{N^s}\}$ are computationally indistinguishable following from the DCR assumption [11], which implies the hidden projective tag property.

Second Construction. This is a second DCR-based ABO dual projective hashing which follows the matrix approach [22]. When instantiated with our generic transformation, this yields a DCR-based $(m, m - \log |\phi(N)|)$-ABO lossy trapdoor functions, which is less efficient than [13]. In order to construct a DCR-based deterministic encryption scheme, we still need a universal hash function that is also universal one-way. The projective map in the following construction already satisfies this, and we will discuss more about it after the construction. Combining the instantiation of DCR-based dual projective hashing [25, Second Construction] with the above instantiation and our generic transformation, this yields a new DCR-based PRIV-CCA secure deterministic encryption, which is less efficient than [7].

- $\mathsf{Setup}(1^\kappa)$: choose a Blum integer $N := PQ$ as above. Pick $\mathbf{p} \leftarrow_\$ \mathbb{Z}_N^m, g \leftarrow_\$ \mathbb{Z}_{N^{s+1}}^*$. Set $hp := (N, (g^{N^s})^{\mathbf{p}})$, $msk := (g, \mathbf{p}, P, Q)$, $\mathcal{X} := \{0,1\}^m$, $\mathcal{Y} := (\mathbb{Z}_{N^{s+1}}^*)^m$, $\mathcal{P} \subseteq \mathbb{Z}_{N^{s+1}}^*$, $\mathcal{U} := (\mathbb{Z}_{N^{s+1}}^*)^{m \times m}$, $\mathcal{W} := \mathbb{Z}_{N^s+1}^m$, $\text{TAG} := \{0, \ldots, 2^{\kappa/2-1}\}$. The projective map α is defined by

$$\alpha((g^{N^s})^{\mathbf{p}}, \mathbf{x}) := (g^{N^s})^{\mathbf{p}^\top \mathbf{x}} \in \mathbb{Z}_{N^{s+1}}^* \text{ where } \mathbf{x} \in \{0,1\}^m, \mathbf{p} \in \mathbb{Z}_N^m.$$

- $\mathsf{Keygen}(msk, b^*)$: choose $\mathbf{w} \leftarrow_\$ \mathbb{Z}_{N^{s+1}}^m$, compute public parameter $\mathbf{U} := (1+N)^{-b^*\mathbf{I}_m} \cdot (g^{N^s})^{\mathbf{w}\mathbf{p}^\top}$. The witness is $\mathbf{w}$. The inversion trapdoor is (P, Q, b^*)
- $\mathsf{Priv}(\mathbf{U}, b, \mathbf{x})$: compute $\Lambda_{\mathbf{U}}(b, \mathbf{x}) := ((1+N)^{b\mathbf{I}_m} \cdot \mathbf{U})^{\mathbf{x}}$.
- $\mathsf{Pub}(\mathbf{U}, b^*, (g^{N^s})^{\mathbf{p}^\top \mathbf{x}}, \mathbf{w})$: compute $((g^{N^s})^{\mathbf{p}^\top \mathbf{x}})^{\mathbf{w}}$.
- $\mathsf{Tdinv}((P, Q, b^*), b, (g^{N^s})^{\mathbf{p}^\top \mathbf{x}}, \Lambda_{\mathbf{U}}(b, \mathbf{x}))$: observe that $\Lambda_{\mathbf{U}}(b, \mathbf{x}) = ((1+N)^{b\mathbf{I}_m} \cdot \mathbf{U})^{\mathbf{x}} = (1+N)^{(b-b^*)\mathbf{x}\mathbf{I}_m} \cdot (g^{N^s})^{\mathbf{w}\mathbf{p}^\top \mathbf{x}} = (1+N)^{(b-b^*)\mathbf{x}} \cdot (g^{N^s})^{\mathbf{w}\mathbf{p}^\top \mathbf{x}}$. Given the inversion trapdoor (i.e., the factorization of N and the projective tag b^*), we can efficiently compute $(b-b^*)\mathbf{x}$. In addition, the restriction $b, b^* \in \{0, \ldots, 2^{\kappa/2}-1\}$ implies that $(b-b^*)$ is smaller than both P and Q and is therefore relatively prime to N. Thus, we can recover $\mathbf{x}$ by computing $(b-b^*)\mathbf{x}(b-b^*)^{-1} \mod N^s$.

Projectiveness: When $\mathbf{U} = (1+N)^{-b^*\mathbf{I}_m} \cdot (g^{N^s})^{\mathbf{w}\mathbf{p}^\top}$ and $b = b^*$, we have

$$\begin{aligned}\mathsf{Priv}(\mathbf{U}, b^*, x) &= ((1+N)^{b^*\mathbf{I}_m} \cdot \mathbf{U})^{\mathbf{x}} = ((1+N)^{b^*\mathbf{I}_m} \cdot (1+N)^{-b^*\mathbf{I}_m} \cdot (g^{N^s})^{\mathbf{w}\mathbf{p}^\top})^{\mathbf{x}} \\ &= ((g^{N^s})^{\mathbf{w}\mathbf{p}^\top})^{\mathbf{x}} = ((g^{N^s})^{\mathbf{p}^\top \mathbf{x}})^{\mathbf{w}} = \mathsf{Pub}(\mathbf{U}, b^*, (g^{N^s})^{\mathbf{p}^\top \mathbf{x}}, \mathbf{w}).\end{aligned}$$

The hidden projective tag property follows from the DCR assumption.

Remark 1. The above projective map α satisfies the universal one-way property and almost universal property. The universal one-way property follows from a similar analysis as that in [7]. Next we show it is almost universal. For any $\mathbf{x} \neq \mathbf{x}' \in \{0,1\}^m$ such that $\alpha((g^{N^s})^{\mathbf{p}}, \mathbf{x}) = \alpha((g^{N^s})^{\mathbf{p}}, \mathbf{x}')$, we get $\sum_{i=1}^m p_i x_i \equiv \sum_{i=1}^m p_i x_i' \mod \lambda(N)$, where $\lambda(N)$ is the least common multiple of $P-1$ and $Q-1$. Without loss of generality, we assume that $x_1 - x_1' \neq 0$, then $p_1 \equiv \sum_{i=2}^m p_i(x_i' - x_i) \mod \lambda(N)$. This happens with probability $\lceil N/\lambda(N) \rceil / N \leq 2/\lambda = 1/pq$.

6.3 Instantiations from LWE

We present the LWE-based construction, which is based on lossy trapdoor functions in [22]. For a real parameter $0 < \beta < 1$, we denote by Ψ_β the distribution over $\mathbb{R}/\mathbb{Z}$ of a normal variable with means 0 and standard deviation $\beta/\sqrt{2\pi}$ then reduced modulo 1. Denote by $\bar{\Psi}_\beta$ the discrete distribution over $\mathbb{Z}_q$ of the random variable $\lfloor qX \rceil \mod q$ where the random variable X has distribution Ψ_β. In the following, we consider the standard LWE parameters m, n, q as well as additional parameters $\tilde{n}, p$ such that

$$m = O(n \log q), \qquad \beta = \Theta(1/q), \qquad \tilde{n} = m/\log p, \qquad \text{and} \qquad p \leq q/16m\tilde{n}.$$

In particular, let $\gamma < 1$ be a constant. We will set $q = \Theta(n^{1+1/\gamma})$ and $p = \Theta(n^{1/\gamma})$. When instantiated with our generic transformations, this yields the LWE-based ABO lossy trapdoor functions in [22]. The projective map α in the following is in fact a universal hash function which is collision-resistance [16] (under small integer solution assumption which is implied by LWE). Since collision-resistance implies universal one-way, α is also universal one-way. When combining the LWE-based dual projective hashing in [25] with our generic transformations, we get a PRIV-CCA-secure deterministic encryption based on LWE, which is less efficient than that in [7]. In addition, we can give another construction following from the extended general framework which is similar to [7].

Let $r : \{0,1\}^{\tilde{n}} \to \mathbb{Z}_q^{m \times \tilde{n}}$ be a function mapping a branch value to its encoded matrix over $\mathbb{Z}_q$ (see [22, Section 6.4]).

- $\mathsf{Setup}(1^\kappa)$: pick $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{n \times m}$. Set $hp := (\mathbf{A})$, $msk := \perp$, $\mathcal{X} := \{0,1\}^m$, $\mathcal{Y} := \mathbb{Z}_q^{\tilde{n}}$, $\mathcal{P} := \mathbb{Z}_q^n$, $\mathcal{U} := \mathbb{Z}_q^{m \times \tilde{n}}$, $\mathcal{W} := \mathbb{Z}_q^{n \times \tilde{n}}$, $\textsc{tag} := \{0,1\}^{\tilde{n}}$. The projective map α is defined by $\alpha(\mathbf{A}, \mathbf{x}) := \mathbf{A}\mathbf{x} \in \mathbb{Z}_q^n$ with $\mathbf{x} \in \{0,1\}^m$.
- $\mathsf{Keygen}(msk, \mathbf{v}^*)$: choose $\mathbf{S} \leftarrow_\$ \mathbb{Z}_q^{n \times \tilde{n}}, \mathbf{E} \leftarrow_\$ (\bar{\Psi}_\beta)^{m \times \tilde{n}}$. Compute public parameter $\mathbf{U} := \mathbf{A}^\top \mathbf{S} + \mathbf{E} - r(\mathbf{v}^*)$. The witness is $\mathbf{S}$. The inversion trapdoor is $(\mathbf{S}, \mathbf{v}^*)$.
- $\mathsf{Priv}(\mathbf{U}, \mathbf{v}, \mathbf{x})$: compute $\Lambda_{\mathbf{U}}(\mathbf{v}, \mathbf{x}) := \mathbf{x}^\top \mathbf{U} + \mathbf{x}^\top r(\mathbf{v}) \in \mathbb{Z}_q^{\tilde{n}}$.
- $\mathsf{Pub}(\mathbf{U}, \mathbf{v}^*, \mathbf{A}\mathbf{x}, \mathbf{S})$: compute $(\mathbf{A}\mathbf{x})^\top \mathbf{S}$.
- $\mathsf{Tdinv}((\mathbf{S}, \mathbf{v}^*), \mathbf{v}, \alpha(\mathbf{A}, \mathbf{x}), \Lambda_{\mathbf{U}}(\mathbf{v}, \mathbf{x}))$: observe that

$$\Lambda_{\mathbf{U}}(\mathbf{v}, \mathbf{x}) = \mathbf{x}^\top \mathbf{U} + \mathbf{x}^\top r(\mathbf{v}) = (\mathbf{A}\mathbf{x})^\top \mathbf{S} + \mathbf{x}^\top \mathbf{E} + \mathbf{x}^\top (r(\mathbf{v} - \mathbf{v}^*)).$$

  Given the inversion trapdoor $(\mathbf{S}, \mathbf{v}^*)$, we can recover $\mathbf{x}^\top \mathbf{E} + \mathbf{x}^\top (r(\mathbf{v}) - r(\mathbf{v}^*))$. The quantity $\mathbf{x}^\top \mathbf{E}$ has small norm, so we can compute x using the bounded-error decoding to recover $\mathbf{x}^\top (r(\mathbf{v}) - r(\mathbf{v}^*))$ and then $\mathbf{x}$.

Projectiveness: The projective property is approximate, that is when $\mathbf{U} := \mathbf{A}^\top \mathbf{S} + \mathbf{E} - r(\mathbf{v}^*)$ and $\mathbf{v} = \mathbf{v}^*$, we have

$$\begin{aligned} \mathsf{Priv}(\mathbf{U}, \mathbf{v}^*, \mathbf{x}) =& \mathbf{x}^\top \mathbf{U} + \mathbf{x}^\top r(\mathbf{v}^*) \\ =& \mathbf{x}^\top (\mathbf{A}^\top \mathbf{S} + \mathbf{E} - r(\mathbf{v}^*)) + \mathbf{x}^\top r(\mathbf{v}^*) \\ =& (\mathbf{A}\mathbf{x})^\top \mathbf{S} + \mathbf{x}^\top \mathbf{E} \approx (\mathbf{A}\mathbf{x})^\top \mathbf{S} = \mathsf{Pub}(\mathbf{U}, \mathbf{v}^*, \mathbf{A}\mathbf{x}, \mathbf{S}). \end{aligned}$$

In fact, for all $\mathbf{x} \in \{0,1\}^m$, with overwhelming probability over $\mathbf{E}$, we have $\mathbf{x}^\top \mathbf{E} \subset [q/p]^{\tilde{n}}$. That is, the projective property holds up to an addictive error term in $[q/p]^{\tilde{n}}$.

The hidden projective tag property follows from the LWE assumption.

ABO Lossy Trapdoor Function. In the lossy mode, we bound the size of the image by $|\mathrm{Img}_\alpha| \cdot (q/p)^{\tilde{n}}$, where $(q/p)^{\tilde{n}}$ accounts for the error incurred by the approximate projective property, then, the lossiness is given by $m - (n \log q + \frac{m}{\log p} \log(\frac{q}{p})) = (1 - \gamma)m - n \log q$.

Acknowledgments. Zongyang Zhang is an International Research Fellow of JSPS and his work is in part supported by NSFC under grant No. 61303201. He thanks Shota Yamada for the discussion of lattice-based construction.

Yu Chen is supported by NSFC under grant No. 61303257 and IIE's Cryptography Research Project under Grant No. Y3Z0011102.

Sherman S. M. Chow is supported by the Early Career Scheme and the Early Career Award of the Research Grants Council, Hong Kong SAR (CUHK 439713), and Direct Grant (4055018) of the Chinese University of Hong Kong.

Zhenfu Cao is supported by NSFC under Nos. 61033014, 61161140320, 61371083 and by the Specialized Research Fund for the Doctoral Program of Higher Education under Grant No. 20130073130004.

Yunlei Zhao is supported by NSFC under Grant No.61272012.

References

1. Abdalla, M., Chevalier, C., Pointcheval, D.: Smooth Projective Hashing for Conditionally Extractable Commitments. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 671–689. Springer, Heidelberg (2009)
2. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and Efficiently Searchable Encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007)
3. Bellare, M., Brakerski, Z., Naor, M., Ristenpart, T., Segev, G., Shacham, H., Yilek, S.: Hedged Public-Key Encryption: How to Protect against Bad Randomness. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 232–249. Springer, Heidelberg (2009)
4. Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic Encryption: Definitional Equivalences and Constructions without Random Oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 360–378. Springer, Heidelberg (2008)
5. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and Impossibility Results for Encryption and Commitment Secure under Selective Opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009)
6. Blazy, O., Pointcheval, D., Vergnaud, D.: Round-Optimal Privacy-Preserving Protocols with Smooth Projective Hash Functions. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 94–111. Springer, Heidelberg (2012)
7. Boldyreva, A., Fehr, S., O'Neill, A.: On Notions of Security for Deterministic Encryption, and Efficient Constructions without Random Oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (2008)
8. Brakerski, Z., Segev, G.: Better Security for Deterministic Public-Key Encryption: The Auxiliary-Input Setting. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 543–560. Springer, Heidelberg (2011)
9. Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
10. Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
11. Damgård, I., Jurik, M., Nielsen, J.B.: A generalization of Paillier's public-key system with applications to electronic voting. Int. J. Inf. Sec. 9(6), 371–385 (2010)

12. Dolev, D., Dwork, C., Naor, M.: Nonmalleable Cryptography. SIAM J. Comput. 30(2), 391–437 (2000)
13. Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More Constructions of Lossy and Correlation-Secure Trapdoor Functions. J. Cryptology 26(1), 39–74 (2013)
14. Fuller, B., O'Neill, A., Reyzin, L.: A Unified Approach to Deterministic Encryption: New Constructions and a Connection to Computational Entropy. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 582–599. Springer, Heidelberg (2012)
15. Gennaro, R., Lindell, Y.: A framework for password-based authenticated key exchange. ACM Trans. Inf. Syst. Secur. 9(2), 181–234 (2006)
16. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Dwork, C. (ed.) STOC, pp. 197–206. ACM (2008)
17. Halevi, S., Kalai, Y.T.: Smooth Projective Hashing and Two-Message Oblivious Transfer. J. Cryptology 25(1), 158–193 (2012)
18. Joye, M., Libert, B.: Efficient Cryptosystems from 2^k-th Power Residue Symbols. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 76–92. Springer, Heidelberg (2013)
19. Katz, J., Vaikuntanathan, V.: Smooth Projective Hashing and Password-Based Authenticated Key Exchange from Lattices. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 636–652. Springer, Heidelberg (2009)
20. Mironov, I., Pandey, O., Reingold, O., Segev, G.: Incremental Deterministic Public-Key Encryption. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 628–644. Springer, Heidelberg (2012)
21. Naor, M., Segev, G.: Public-Key Cryptosystems Resilient to Key Leakage. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 18–35. Springer, Heidelberg (2009)
22. Peikert, C., Waters, B.: Lossy Trapdoor Functions and Their Applications. SIAM J. Comput. 40(6), 1803–1844 (2011)
23. Raghunathan, A., Segev, G., Vadhan, S.P.: Deterministic Public-Key Encryption for Adaptively Chosen Plaintext Distributions. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 93–110. Springer, Heidelberg (2013)
24. Wee, H.: Efficient Chosen-Ciphertext Security via Extractable Hash Proofs. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 314–332. Springer, Heidelberg (2010)
25. Wee, H.: Dual Projective Hashing and Its Applications - Lossy Trapdoor Functions and More. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 246–262. Springer, Heidelberg (2012)
26. Wichs, D.: Barriers in cryptography with weak, correlated and leaky sources. In: Kleinberg, R.D. (ed.) ITCS, pp. 111–126. ACM (2013)