Identity-Based Extractable Hash Proofs and Their Applications

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

[1] State Key Laboratory of Information Security (SKLOIS),
Institute of Information Engineering, Chinese Academy of Sciences, 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 paper, we introduce a general paradigm called identity-based extractable hash proof system (IB-EHPS), which is an extension of extractable hash proof system (EHPS) proposed by Wee (CRYPTO '10). We show how to construct identity-based encryption (IBE) scheme from IB-EHPS in a simple and modular fashion. Our construction provides a generic method of building and interpreting CCA-secure IBE schemes based on computational assumptions. As instantiations, we realize IB-EHPS from the bilinear Diffie-Hellman assumption and the modified bilinear Diffie-Hellman assumption, respectively.

1 Introduction

Security against adaptive chosen-ciphertext attack (CCA-security) [27] is now accepted as the standard security notion for public-key encryption (PKE) schemes as well as identity-based encryption (IBE) schemes. In contrast to security against adaptive chosen-plaintext attack (CPA-security) [25], CCA-security captures the immunity against an active adversary who is given access to a decryption oracle that allows it to obtain the decryptions of ciphertexts of its choice.

On the other hand, in most cases related to cryptography, decisional assumptions form a much stronger class of assumptions than the corresponding search (computational) assumptions[1]. As such, cryptosystems based on search problems are generally preferred to those based on decisional assumptions. From now on, we will use the term *computational* and *search* interchangeablely.

Up to now, only a handful of IBE schemes [11, 14, 19] have been proven to be CCA-secure from computational assumptions in the standard model. Besides, there seems no overarching concept explaining these constructions. Inspired by the notion of extractable hash proof system [31] in the public key setting, we introduce a new notion named identity-based extractable hash proof system and show how to construct CCA-secure IBE schemes from it.

[⋆] Corresponding author.

[1] Unless the decisional assumption can be proved equivalent to its computational counterpart, as it is the case with cryptosystems based on the problem of "leaning with error" (LWE) [26].

F. Bao, P. Samarati, and J. Zhou (Eds.): ACNS 2012, LNCS 7341, pp. 153–170, 2012.

1.1 Background

The concept of identity-based encryption (IBE) was introduced by Shamir [28] in 1984. Boneh and Franklin [6] proposed the first practical IBE scheme whose security is based on the computational bilinear Diffie-Hellman (CBDH) assumption. Cocks [12] described another IBE scheme based on the decisional quadratic residues (DQR) assumption modulo a composite. Both of them are proven secure under the random oracle model [2]. However, a proof in the random oracle model can only serve as a heuristic argument and possibly lead to insecure schemes in the standard model. This posed an interesting problem of constructing IBE schemes in the standard model.

First, Canetti, Halevi, and Katz [8] made the breakthrough by giving a solution in the standard model, but under a weaker notion named "selective-identity" where the attacker must declare the target identity id^* before seeing the public parameters. Boneh and Boyen [4] then provided two efficient selective-identity CPA-secure IBE schemes known as BB_1-IBE and BB_2-IBE. The former is based on the decisional bilinear Diffie-Hellman (DBDH) assumption while the latter is based the decisional q-BDHI assumption. Subsequently, Waters [29] proposed an efficient and adaptive-identity CPA-secure IBE scheme (Waters-IBE) in the standard model which is also based on the DBDH assumption by employing Waters hash in place of Boneh-Boyen hash used in BB_1-IBE. One drawback is that it suffers from large public parameter size. Gentry [15] proposed an IBE scheme (Gentry-IBE) which enjoys short public parameters and tight security reduction. Although Gentry-IBE achieves adaptive-identity CCA-security in the standard model, it did so at the cost of relying a non-standard and non-static assumption called the decisional q-ABHDE assumption. Waters [30] then introduced dual system encryption methodology and proposed an adaptive-identity CPA-secure IBE scheme based on the DBDH assumption and the decisional linear (DLIN) assumption in the standard model. Recently, Gentry *et al.* [16] proposed an IBE scheme based on the LWE assumption in the random oracle model. Cash *et al.* [9] and Agrawal *et al.* [1] showed how to construct IBE schemes based on the LWE assumption in the standard model.

As previously stated, CCA-security is the *de facto* level of security required for IBE schemes used in practice. Unfortunately, constructing CCA-secure IBE scheme without resorting to random oracle heuristic turns out to be difficult. Boneh, Canetti, Halevi, and Katz [5] proposed a generic transformation (known as the BCHK transformation) from any CPA-secure 2-level HIBE scheme to a CCA-secure IBE scheme, which is the only generic approach known for constructing efficient CCA-secure IBE in the standard model.

1.2 Motivation

As we have already mentioned, a decisional assumption is generally stronger than its computational counterpart. From both theoretical and practical perspective, it is more desirable to reduce the security of cryptographic schemes to computational assumptions. Considering an IBE scheme obtained from the BCHK transformation, its CCA-security relies on the CPA-security of the underlying 2-level HIBE scheme and the security of one-time signature or MAC. Hence its assumption cannot be directly counted as computational or decisional assumption. However, the indistinguishability against CPA-attack is of decisional flavor, thus it is arguably closer to decisional assumptions.

Haralambiev *et al.* [19] proposed several efficient PKE schemes in the standard model. They also sketched that one of their PKE schemes can be extended to a BB_1-style identity-based key encapsulation mechanism (IB-KEM). Galindo [14] gave an IB-KEM from the PKE scheme due to Hanaoka and Kurosawa [18]. Chen *et al.* [11] proposed another BB_1-style IB-KEM. All the above IB-KEMs are proven to be selective-identity CCA-secure based on the CBDH assumption in the standard model. All of them fall outside of the BCHK [5] methodology. While the IB-KEMs due to [19] and [11] are similar, it seems that the IB-KEM [14] relies on different techniques to achieve CCA-security. So far, there is no overarching framework explaining these constructions.

Recently, several CCA-secure PKE schemes from various computational assumptions emerged, such as [10, 18–20]. Inspired in part by hash proof system (HPS) [13], Wee [31] introduced the notion of extractable hash proof system (EHPS) and showed how to derive efficient CCA-secure PKE via EHPS. Roughly speaking, EHPS resembles hash proof system (HPS) [13] in that both of them are essentially a special kind of non-interactive zero-knowledge proof, except that EHPS replaces the soundness requirement with a *proof of knowledge property* [27]. The framework of EHPS does not only encompass a series of CCA-secure PKE schemes [21,22] based on decisional assumptions, but also can explain a series of CCA-secure PKE schemes [19, 20] based on computational assumptions in a unified way, which is the most appealing advantage of EHPS.

Although the realm of IBE and PKE are inherently different, the techniques are sometimes interchangeable. Motivated by the above discussion, we find the following intriguing question:

Does there exist a general framework for the construction of identity-based encryption from computational assumptions in the standard model?

1.3 Our Contributions

EHPSs and their benefits are confined to the realm of public-key setting. In this paper we bring them to the identity-based setting, defining identity-based extractable hash proof system (IB-EHPS). Using IB-EHPS, we obtain new insights into the construction of CCA-secure IBE schemes. In particular, we show that this notion unifies many seemingly unrelated IBE constructions under a single framework. We summarize our main contributions as follows.

Identity-Based Extractable Hash Proof Systems. We introduce the notion of IB-EHPS by tailoring EHPS to the identity-based setting. We show that IB-EHPS instantly yields adaptive-identity CPA-secure IBE. However, the basic IB-EHPS is too generic to encompass more applications. To resolve this problem, we further propose the notion of all-but-one (ABO) IB-EHPS, which can in turn be used to construct adaptive-identity CCA-secure IBE.

Practical CCA-secure IBE from IB-EHPS. We present two ABO IB-EHPSs from the CBDH assumption and the modified CBDH assumption, respectively. As a result, we obtain two efficient adaptive-identity CCA-secure IBE schemes based on computational assumptions in the standard model.

2 Preliminaries

2.1 Definitions

For a positive integer n, we use $[n]$ to denote the set $[n] = \{1, \dots, n\}$. For a finite set X, we use $x \overset{R}{\leftarrow} X$ to denote that x is sampled from X uniformly at random. The main security parameter through this paper is κ, and all algorithms are implicitly given κ as input. We use standard asymptotic notation O and o to denote the growth of functions. Let $\mathsf{poly}(\kappa)$ denote an unspecified function $f(\kappa) = O(\kappa^c)$ for some constant c. Let $\mathsf{negl}(\kappa)$ denote an unspecified function $f(\kappa)$ such that $f = o(\kappa^{-c})$ for every constant c. We say that a probability is overwhelming if it is $1 - \mathsf{negl}(\kappa)$. A probabilistic polynomial-time (PPT) algorithm is a randomized algorithm that runs in time $\mathsf{poly}(\kappa)$. If $\mathcal{A}$ is a randomized algorithm, we write $z \leftarrow \mathcal{A}(x_1, \dots, x_n; r)$ to indicate that $\mathcal{A}$ outputs z on inputs $(x_1, \dots, x_n)$ and random coins r. We will omit r and write $z \leftarrow \mathcal{A}(x_1, \dots, x_n)$ when it is not necessary to make explicit the randomness $\mathcal{A}$ uses. We assume that an algorithm returns $\perp$ if any of its inputs is $\perp$.

2.2 Identity-Based Key Encapsulation Mechanisms

Instead of providing the full functionality of an IBE scheme, in many applications it is sufficient to allow sender and receiver to agree on a common random session key. This can be accomplished by *identity-based key encapsulation mechanism* (IB-KEM) as formalized in [3]. Considering there are many practical reasons to prefer an IB-KEM over an IBE scheme, we define IBE schemes as IB-KEM in this paper. An IB-KEM consists of four PPT algorithms as follows:

- $\mathsf{Setup}(\kappa)$: takes as input a security parameter κ, outputs the master public key mpk and the master secret key msk. mpk will be used as an implicit input by all other algorithms KeyGen, Encap, Decap. Let I, C, and K be the identity space, ciphertext space, and the key space (for DEM), respectively.
- $\mathsf{KeyGen}(msk, id)$: takes as input msk and an identity $id \in I$, outputs a private key sk of id.
- $\mathsf{Encap}(id)$: takes as input an identity $id \in I$, outputs a ciphertext $c \in C$ and a DEM key $k \in K$.
- $\mathsf{Decap}(sk, c)$: takes as input a private key sk of identity id and a ciphertext c, outputs a DEM key $k \in K$ or an distinguished symbol $\perp$ (which is not in K) indicating that c is not consistent under id. Here we say that a ciphertext is *consistent* or *well-formed* or *valid* if it can be "honestly generated" by the encryption algorithm. For a PKE or IBE scheme, if anyone can do the "consistency check", we say that it is *public verifiable*. Otherwise, we say that it is *private verifiable*.

We refer to [23] for formal security definition of IB-KEM. For correctness, we require that for any $(mpk, msk) \leftarrow \mathsf{Setup}(\kappa)$, any $(c, k) \leftarrow \mathsf{Encap}(id)$, and any $sk \leftarrow \mathsf{KeyGen}(msk, id)$, we have $\Pr[\mathsf{Decap}(sk, c) = k] = 1$.

2.3 Bilinear Diffie-Hellman Assumption

Let $(p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathsf{GroupGen}(1^\kappa)$, where $\mathsf{GroupGen}(\cdot)$ is a bilinear groups parameters generator [7]. Let g be a random generator of $\mathbb{G}$. Define $\mathrm{bdh}(A, B, C) := T$, where $A = g^a$, $B = g^b$, $C = g^c$, and $T = e(g, g)^{abc}$. The computational bilinear Diffie-Hellman (CBDH) problem is computing $\mathrm{bdh}(A, B, C)$ given random $A, B, C \in \mathbb{G}$. The CBDH assumption asserts that the CBDH problem is hard, that is, $\Pr[\mathcal{A}(A, B, C) = \mathrm{bdh}(A, B, C)] \leq \mathsf{negl}(\kappa)$ for all PPT algorithms $\mathcal{A}$.

In the bilinear setting, the Goldreich-Levin theorem [17] gives us the following lemma for a Goldreich-Levin hardcore predicate $f_{\mathrm{gl}} : \mathbb{G}_T \times \{0, 1\}^u \rightarrow \{0, 1\}$.

Lemma 2.1 *Let $A, B, C \xleftarrow{R} \mathbb{G}$, $R \xleftarrow{R} \{0, 1\}^u$, $K = f_{\mathrm{gl}}(\mathrm{bdh}(A, B, C), R)$, and $U \xleftarrow{R} \{0, 1\}$. Suppose there exists a PPT algorithm $\mathcal{B}$ distinguishing the distributions $\Delta_{\mathrm{bdh}} = (g, A, B, C, K, R)$ and $\Delta_{\mathrm{rand}} = (g, A, B, C, U, R)$ with non-negligible advantage. Then there exists a PPT algorithm solving the CBDH problem with non-negligible correct probability.*

The modified computational bilinear Diffie-Hellman (mCBDH) problem [24] is similar to the CBDH problem except that an additional point $B' = g^{b^2}$ is given. We can prove a similar lemma regarding mCBDH problem as Lemma 2.1.

2.4 Binary Relations for Search Problems

A search problem $\mathbf{S} = (S_\kappa)_{\kappa \geq 0}$ is a collection of distributions. For every value of $\kappa \geq 0$, an instance of S_κ specifies two finite, non-empty sets X and W, public parameter PP, and a binary relation $\mathsf{R}_{\mathrm{pp}} \in X \times W$. A search problem also provides two algorithms, namely SampS and SampR. SampS takes as input a security parameter κ, and outputs an instance of S_κ. We write $(X, W, \mathrm{PP}, \mathsf{R}_{\mathrm{pp}}) \leftarrow \mathsf{SampS}(\kappa; \mathrm{SP})$, where SP is the random coins used in SampS. SampR takes as input PP, and outputs a tuple (x, w) belong to R_{pp}. We write $(x, w) \leftarrow \mathsf{SampR}(\mathrm{PP}; r)$, where r is the random coins used in SampR. Note that PP is often assumed to be an implicit input and it is useful to make the random coins explicitly in SampR algorithm, thus we often write $\mathsf{SampR}(r)$ henceforth whenever the context is clear. Different to the requirement in EHPS [31], we do not require that R_{pp} can be efficiently verifiable in IB-EHPS.

Intuitively, the relation R_{pp} corresponds to a hard search problem, that is, given a random element $x \in X$, it is hard to find $w \in W$ such $(x, w) \in \mathsf{R}_{\mathrm{pp}}$. More formally, we say that a binary relation R_{pp} is one-way if:

- with overwhelming probability over PP, for any $x \in X$, there exists at most one $w \in W$ such that $(x, w) \in \mathsf{R}_{\mathrm{pp}}$ (we say that w is a *witness* for x); and
- there is an efficiently computable function F from W to $\{0, 1\}^l$ for some positive integer l such that given x, $\mathsf{F}(w)$ is pseudo-random over $\{0, 1\}^l$ where $(x, w) \leftarrow \mathsf{SampR}(\mathrm{PP})$.

For relations where computing w given x is hard on average, we may derive a function GL with a one-bit output via the Goldreich-Levin hardcore predicate f_{gl}. Note that GL is an instantiation of the above function F.

Bilinear Diffie-Hellman Relation. Let $(p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathsf{GroupGen}(\kappa)$. The public parameter PP is given by (g, g^a, g^b) for a random $g \in \mathbb{G}$ and random $a, b \overset{R}{\leftarrow} \mathbb{Z}_p$[2]. We consider the bilinear Diffie-Hellman relation over $\mathbb{G} \times \mathbb{G}_T$:

$$\mathsf{R}_{\text{pp}}^{\mathsf{bdh}} = \left\{(x, w) \in \mathbb{G} \times \mathbb{G}_T : w = e(g, x)^{ab}\right\}$$

The associated SampR picks $r \overset{R}{\leftarrow} \mathbb{Z}_p$ and outputs $(g^r, e(g^a, g^b)^r)$. Lemma 2.1 shows that we may extract a single hardcore bit from w using $\mathsf{GL}(w)$ for relation $\mathsf{R}_{\text{pp}}^{\mathsf{bdh}}$. The modified BDH relation $\mathsf{R}_{\text{pp}}^{\mathsf{mbdh}}$ can be defined analogously.

2.5 General Hashing

Let X, I, and Y be finite, non-empty sets. Let $\mathsf{H} = (\mathsf{H}_{mpk})_{mpk \in MPK}$ be a collection of functions indexed by MPK, so that for every $mpk \in MPK$, H_{mpk} is a function from $I \times X$ into Y. We call $\mathbf{H} = (\mathsf{H}, MPK, I, X, Y)$ a hash family.

3 Identity-Based Extractable Hash Proofs

An IB-EHPS $\mathbf{P}$ for $\mathbf{S}$ associating with each instance $(X, W, \text{PP}, \mathsf{R}_{\text{pp}}) \leftarrow \mathsf{SampS}(\kappa)$ of S_κ and an identity space I and a hash family $\mathbf{H} = (\mathsf{H}, MPK, I, X, Y)$, is a tuple of algorithms ($\mathsf{SetupExt}$, $\mathsf{SetupHash}$, KeyGen, KeyGen^*, Pub, Priv, Ext). Loosely speaking, an IB-EHPS can behave in one of two modes, namely the extraction mode and the hashing mode. We will rely on the extraction mode for the normal functionality of the resulting IBE scheme, and on the hashing mode for the proof of security.

Extraction Mode

- $\mathsf{SetupExt}(\text{PP}, \text{SP})$: takes as input (PP, SP), outputs the master public key mpk and the master secret key msk.
- $\mathsf{Pub}(mpk, id, r)$: takes as input mpk, an identity $id \in I$ and random coins r, outputs $y \in Y$ such that $y = \mathsf{H}_{mpk}(id, x)$ where $(x, w) \leftarrow \mathsf{SampR}(r)$. This is the *public evaluation algorithm.*
- $\mathsf{KeyGen}(msk, id)$: takes as input msk and an identity $id \in I$, outputs a private key sk for id.
- $\mathsf{Ext}(sk, x, y)$: takes as input a private key sk of identity $id \in I$, $x \in X$ and $y \in Y$, outputs $w \in W$.

For the correctness of extraction mode, we require that for any $(mpk, msk) \leftarrow \mathsf{SetupExt}(\text{PP}, \text{SP})$ and any $id \in I$ and any $sk \leftarrow \mathsf{KeyGen}(msk, id)$, we have $y = \mathsf{H}_{mpk}(id, x) \Longrightarrow (x, \mathsf{Ext}(sk, x, y)) \in \mathsf{R}_{\text{pp}}$.

[2] We assume PP also includes p and the descriptions of $(e, \mathbb{G}, \mathbb{G}_T)$.

Hashing Mode

- $\mathsf{SetupHash}(\mathrm{PP})$: takes PP as input, outputs the master public key mpk and the master secret key msk^*. msk^* implicitly splits the whole identity space I into two orthogonal subspaces I_1 and I_2, namely $I = I_1 \cup I_2$ and $I_1 \cap I_2 = \emptyset$.
- $\mathsf{Priv}(msk^*, id, x)$: takes as input msk^* and an identity $id \in I$, if $id \in I_2$ outputs $y \in Y$, else outputs $\perp$. This is the *private evaluation algorithm.*
- $\mathsf{KeyGen}^*(msk^*, id)$: takes as input msk^* and an identity $id \in I$, if $id \in I_1$ outputs a private key sk for id, else outputs $\perp$.

For the correctness of hashing mode, we require that for any $(mpk, msk) \leftarrow \mathsf{SetupHash}(\mathrm{PP})$ and any $id \in I_2$, we have $\mathsf{Priv}(msk^*, id, x) = \mathsf{H}_{mpk}(id, x)$.

Indistinguishability. We require the first output (mpk) of $\mathsf{SetupExt}(\mathrm{PP}, \mathrm{SP})$ and $\mathsf{SetupHash}(\mathrm{PP})$ are statistically indistinguishable. For any mpk and any identity $id \in I_1$, we require the output of $\mathsf{KeyGen}(msk, id)$ and $\mathsf{KeyGen}^*(msk^*, id)$ are statistically indistinguishable.

Well Partition. We now set a property that is sufficient for the existence of an efficient transformation that we will use to obtain CPA-secure IB-KEM. Intuitively, this property guarantees that no PPT adversary can distinguish the CPA-security games simulated by operating IB-EHPS in extraction mode and hashing mode with non-negligible probability. We formally define this property via the following game played between a PPT adversary $\mathcal{A}$ and a challenger $\mathcal{CH}$.

Given PP and SP, $\mathcal{CH}$ picks $b \xleftarrow{R} \{0, 1\}$, and plays Sub-Game b with $\mathcal{A}$.

Sub-Game 0. $\mathcal{CH}$ interacts with $\mathcal{A}$ by operating IB-EHPS in extraction mode.
Setup: $\mathcal{CH}$ generates $(mpk, msk) \leftarrow \mathsf{SetupExt}(\mathrm{PP}, \mathrm{SP})$ and gives mpk to $\mathcal{A}$. $\mathcal{CH}$ also samples $(x^*, w^*) \leftarrow \mathsf{SampR}(r^*)$ and records them for latter use.
Phase 1 - Private key queries: When $\mathcal{A}$ submits an private key query $\langle id \rangle$, $\mathcal{CH}$ responds with $\mathsf{KeyGen}(msk, id)$.
Phase Middle: When $\mathcal{A}$ submits an identity $id^* \in I$ on the condition that id^* did not appear in any private key query in Phase 1, $\mathcal{CH}$ obtains $y^* = \mathsf{H}_{mpk}(id^*, x^*)$ by evaluating $\mathsf{Pub}(mpk, id^*, r^*)$, then sets $k_0^* = \mathsf{F}(w^*)$ and $k_1^* \xleftarrow{R} \{0, 1\}^l$. $\mathcal{CH}$ picks a random bit $\beta \in \{0, 1\}$ and returns (x^*, y^*, k_β^*) to $\mathcal{A}$.
Phase 2 - Private key queries: Same as Phase 1 except that the private key query $\langle id^* \rangle$ is not allowed.

Sub-Game 1. $\mathcal{CH}$ interacts with $\mathcal{A}$ by operating IB-EHPS in hashing mode.
Setup: $\mathcal{CH}$ generates $(mpk, msk^*) \leftarrow \mathsf{SetupHash}(\mathrm{PP})$ and gives mpk to $\mathcal{A}$. $\mathcal{CH}$ also samples $(x^*, w^*) \leftarrow \mathsf{SampR}(r^*)$ and records them for latter use.
Phase 1 - Private key queries: When $\mathcal{A}$ submits a private key query $\langle id \rangle$, $\mathcal{CH}$ responds with $\mathsf{KeyGen}(msk^*, id)$.
Phase Middle: When $\mathcal{A}$ submits an identity $id^* \in I$ on the condition that id^* did not appear in any private key query in Phase 1, $\mathcal{CH}$ computes $y^* = \mathsf{H}_{mpk}(id, x^*)$ via $\mathsf{Priv}(msk^*, id^*, x^*)$, and sets $k_0^* = \mathsf{F}(w^*)$ and $k_1^* \xleftarrow{R} \{0, 1\}^l$. $\mathcal{CH}$ picks a random bit $\beta \in \{0, 1\}$ and returns (x^*, y^*, k_β^*) to $\mathcal{A}$.

Phase 2 - Private key queries: Same as Phase 1 except that the private key query $\langle id^* \rangle$ is not allowed.

At the end of the game, $\mathcal{A}$ outputs its guess b' for b and wins the game if $b = b'$. Let $\Pr[\mathcal{A} \text{ wins}]$ be the probability that $\mathcal{A}$ wins the game, where the probability space is over the random coins consumed by $\mathcal{CH}$. Let δ be a real number in $[0, 1]$. It is straightforward to see that if $\Pr[\mathcal{A} \text{ wins}] \leq 1 - \frac{1}{2}\delta$, then the probability that the $\mathcal{A}$'s view in Sub-Game 1 is identical to Sub-Game 0 is at least δ. Let Q_e be the number of private key queries. We say such an IB-EHPS is (Q_e, δ)-well-partition.

All-But-One Identity-Based Extractable Hash Proofs. For our applications, it is convenient to work with a richer abstraction. More precisely, an ABO IB-EHPS is a tuple of algorithms $(\mathsf{SetupExt}, \mathsf{SetupABO}, \mathsf{Pub}, \mathsf{Priv}, \mathsf{Verify}, \mathsf{Verify}^*, \mathsf{KeyGen}, \mathsf{KeyGen}^*, \mathsf{Ext}, \mathsf{Ext}^*)$.

Extraction Mode

- The algorithms $\mathsf{SetupExt}$, Pub, and KeyGen related to the extraction mode are identical to that in IB-EHPS.
- $\mathsf{Verify}(id, sk, x, y)$: takes as input an identity $id \in I$, a private key sk for id, $x \in X$ and $y \in Y$, if $y = \mathsf{H}_{mpk}(id, x)$ returns 1, else returns 0. Particularly, when sk is not necessary, we say H_{mpk} is *public verifiable*.
- $\mathsf{Ext}(sk, x, y)$: takes as input a private key sk for identity $id \in I$, $x \in X$ and $y \in Y$, if $\mathsf{Verify}(id, sk, x, y) = 1$ then outputs $w \in W$, else outputs $\perp$.

For the correctness of extraction mode, we require that for any $(mpk, msk) \leftarrow \mathsf{SetupExt}(\mathrm{PP}, \mathrm{SP})$, any $id \in I$ and any $sk \leftarrow \mathsf{KeyGen}(msk, id)$, we have:

$$y = \mathsf{H}_{mpk}(id, x) \Longrightarrow (x, \mathsf{Ext}(sk, x, y)) \in \mathsf{R}_{\mathrm{pp}} \tag{1}$$

ABO Hashing Mode

- $\mathsf{SetupABO}(\mathrm{PP}, x^*)$: similar to $\mathsf{SetupHash}(\mathrm{PP})$ in IB-EHPS except taking an extra input $x^* \in X$.
- $\mathsf{KeyGen}^*(msk^*, id)$: same as KeyGen^* in IB-EHPS.
- $\mathsf{Priv}(msk^*, id, x)$: takes as input msk^*, $id \in I$, and $x \in X$, if $id \in I_2$ and $x = x^*$ outputs $y \in Y$, else outputs $\perp$.
- $\mathsf{Verify}^*(id, msk^*, x, y)$: takes as input an identity $id \in I$, msk^*, $x \in X$ and $y \in Y$, if $y = \mathsf{H}_{mpk}(id, x)$ returns 1 else returns 0. When H_{mpk} is public verifiable, msk^* is not necessary.
- $\mathsf{Ext}^*(msk^*, x, y)$: takes as input msk^*, $x \in X$, and $y \in Y$, if $x \neq x^*$ and $\mathsf{Verify}^*(id, msk^*, x, y) = 1$ then outputs $w \in W$, else outputs $\perp$.

For the correctness of ABO hashing mode, we require for any $x^* \in X$ and any $(mpk, msk^*) \leftarrow \mathsf{SetupABO}(\mathrm{SP}, x^*)$ and any $id \in I_2$, we have $\mathsf{Priv}(msk^*, id, x^*) = \mathsf{H}_{mpk}(id, x^*)$, and for any $id \in I$ if $x \neq x^*$ we have:

$$y = \mathsf{H}_{mpk}(id, x) \Longrightarrow (x, \mathsf{Ext}^*(msk^*, x, y)) \in \mathsf{R}_{\mathrm{pp}} \tag{2}$$

Indistinguishability. We require that the similar indistinguishable properties hold as that for IB-EHPS, namely for any $x^* \in X$ the first output of $\mathsf{SetupExt}(\mathrm{PP}, \mathrm{SP})$ and $\mathsf{SetupHash}(\mathrm{PP}, x^*)$ are statistically indistinguishable. For any mpk and any identity $id \in I_1$, we require that the output of $\mathsf{KeyGen}(msk, id)$ and $\mathsf{KeyGen}^*(msk^*, id)$ are statistically indistinguishable.
Well Partition. This property for ABO IB-EHPS is defined analogously as that for IB-EHPS. We formally defined it via the following game played between a PPT adversary $\mathcal{A}$ and a challenger $\mathcal{CH}$.

Given PP and SP, $\mathcal{CH}$ picks $b \xleftarrow{R} \{0, 1\}$, and plays Sub-Game b with $\mathcal{A}$.

Sub-Game 0. $\mathcal{CH}$ interacts with $\mathcal{A}$ by operating ABO IB-EHPS in extraction mode.
Setup: Same as Sub-Game 0 in IB-EHPS.
Phase 1 - Private key queries: Same as Sub-Game 0 in IB-EHPS.
Phase 1 - Decapsulation queries: When $\mathcal{A}$ submits a query $\langle id, x, y\rangle$, if $x = x^*$, $\mathcal{CH}$ directly returns $\perp$. Otherwise $\mathcal{CH}$ responds with $\mathsf{F}(\mathsf{Ext}(sk, x, y))$.
Phase Middle: Same as Sub-Game 0 in IB-EHPS.
Phase 2 - Private key queries: Same as Sub-Game 0 in IB-EHPS.
Phase 2 - Decapsulation queries: When $\mathcal{A}$ submits a query $\langle id, x, y\rangle$, $\mathcal{CH}$ computes $sk \leftarrow \mathsf{KeyGen}(msk, id)$ and responds with $\mathsf{F}(\mathsf{Ext}(sk, x, y))$. The query $\langle id^*, x^*, y^*\rangle$ is not allowed.

Sub-Game 1. $\mathcal{CH}$ interacts with $\mathcal{A}$ by operating ABO IB-EHPS in ABO hashing mode.
Setup: $\mathcal{CH}$ generates $(mpk, msk^*) \leftarrow \mathsf{SetupABO}(\mathrm{PP}, x^*)$ and gives mpk to $\mathcal{A}$. $\mathcal{CH}$ also samples $(x^*, w^*) \leftarrow \mathsf{SampR}(r^*)$ and records it for latter use.
Phase 1 - Private key queries: Same as Sub-Game 1 in IB-EHPS.
Phase 1 - Decapsulation queries: When $\mathcal{A}$ submits a query $\langle id, x, y\rangle$, if $x = x^*$, $\mathcal{CH}$ returns $\perp$. Otherwise if $id \in I_1$, $\mathcal{CH}$ extracts $sk = \mathsf{KeyGen}^*(msk^*, id)$ and responds with $\mathsf{F}(\mathsf{Ext}(sk, x, y))$, else responds with $\mathsf{F}(\mathsf{Ext}^*(msk^*, x, y))$.
Phase Middle: Same as Sub-Game 1 in IB-EHPS.
Phase 2 - Private key queries: Same as Sub-Game 1 in IB-EHPS.
Phase 2 - Decapsulation queries: When $\mathcal{A}$ submits a query $\langle id, x, y\rangle$, if $id \in I_1$, $\mathcal{CH}$ computes $sk = \mathsf{KeyGen}^*(msk^*, id)$ and responds with $\mathsf{F}(\mathsf{Ext}(sk, x, y))$, else responds with $\mathsf{F}(\mathsf{Ext}^*(msk^*, x, y))$. The extraction query $\langle id^*, x^*, y^*\rangle$ is not allowed.

At the end of the game, $\mathcal{A}$ outputs its guess b' for b and wins the game if $b = b'$. Similar to the analysis we have done before, if $\Pr[\mathcal{A} \text{ wins}] \leq 1 - \frac{1}{2}\delta$, then the probability that the $\mathcal{A}$'s view in Sub-Game 1 is identical to Sub-Game 0 is at least δ. Let Q_e and Q_d be the number of private key queries and extraction queries, respectively. We say such an ABO IB-EHPS is (Q_e, Q_d, δ)-well-partition.

In ABO IB-EHPS, property (1) for the extraction mode ensures the functionality of the resulting IB-KEM while the property (2) for the ABO hashing mode ensures the correctness of simulation. The crux to achieve CCA-security is to make sure that the decryption oracle does not help the adversary to distinguish w^* from random; in other words, the output of the decryption algorithm should contain no knowledge of w^* related to x^* when the input ciphertext (x^*, y) is not consistent. In line of this, to yield CCA-secure IBE, the ABO IB-EHPS should also have the following two properties:

$$y \neq \mathsf{H}_{mpk}(id, x) \Longrightarrow (x, \mathsf{Ext}(sk, x, y)) \notin \mathsf{R}_{\mathrm{pp}} \tag{3}$$

$$y \neq \mathsf{H}_{mpk}(id, x) \Longrightarrow (x, \mathsf{Ext}^*(msk^*, x, y)) \notin \mathsf{R}_{\mathrm{pp}} \tag{4}$$

We achieve properties (3) and (4) by equipping the ABO IB-EHPS with algorithms Verify and Verify^* which can determine if $y = \mathsf{H}_{mpk}(id, x)$, and algorithms Ext and Ext^* returns a distinguished symbol $\perp$ when $y \neq \mathsf{H}_{mpk}(id, x)$. We note that it is also possible to achieve properties (3) and (4) without requiring algorithms Verify and Verify^* available. The trick is for certain relation R we may re-design algorithms Ext and Ext^* smartly using the "implicit rejection" idea [21, 24], namely for $y \neq \mathsf{H}_{mpk}(id, x)$, $\mathsf{Ext}(sk, x, y)$ and $\mathsf{Ext}^*(msk^*, x, y)$ returns a random value $w \in W$ which is independent of x. Thus the properties (3) and (4) will hold with overwhelming probability.

Combining properties (3) and (4) with (1) and (2), the ABO IB-EHPS in fact has the following stronger properties: $y = \mathsf{H}_{mpk}(id, x) \Longleftrightarrow (x, \mathsf{Ext}(sk, x, y)) \in \mathsf{R}_{\mathrm{pp}}$ for the extraction mode and $y = \mathsf{H}_{mpk}(id, x) \Longleftrightarrow (x, \mathsf{Ext}(msk^*, x, y)) \in \mathsf{R}_{\mathrm{pp}}$ for the ABO mode (when $x \neq x^*$), which is reminiscent of ABO EHPS [31]. The key difference is that [31] achieves properties (3) and (4) by requiring the relation R_{pp} can be efficiently verifiable, which may make it too stringent to cover many known CCA-secure IBE schemes, such as [11, 19, 23].

3.1 Relation to Extractable Hash Proof System

IB-EHPS is the corresponding notion of EHPS in the IBE setting. However, we stress that the extension is not straightforward for the following main differences.

1. The (ABO) hashing mode for (ABO) IB-EHPS is defined in partitioning style. More precisely, the setup algorithm generates (mpk, msk^*) and implicitly splits the whole identity space I into two orthogonal subspaces, — 1) I_1: identities for which KeyGen^* can generate private keys; and 2) I_2: identities for which Priv can evaluate the hash value. We note that (ABO) IB-EHPS inherently relies on the partitioning strategy. Suppose that there is an identity id belongs to the intersection of I_1 and I_2, then given (PP, x) one can compute the corresponding w such that $(x, w) \in \mathsf{R}_{\mathrm{pp}}$ by itself as follows: first computes $y = \mathsf{H}_{mpk}(id, x)$ via $\mathsf{Priv}(msk^*, id, x)$, then obtains a private key sk of id via $\mathsf{KeyGen}(msk^*, id)$ and uses it to extract w via $\mathsf{Ext}(sk, x, y)$. This contradicts the one-wayness of R_{pp}. This feature of IB-EHPS makes it particularly well-suited to yield IBE schemes whose provable security follows the partitioning strategy [15, 30].
2. In ABO EHPS, the ABO hashing mode is defined with respect to a tag t^*, which in turn is the hash value of x^* for some target collision resistant (TCR)

hash function. Hence the correctness of the ABO hashing mode is related to the TCR hash function. In our case, we define the ABO hashing mode directly with respect to x^*. We do so out of two reasons. One is that for an abstract paradigm it is more preferable to minimize the dependence on other primitives, while the other is that the proof for the transformation from IB-EHPS to CCA-secure IBE would be rather clean and simple. Nevertheless, TCR hash function turns out to be a useful tool when instantiating EHPS/IB-EHPS from concrete number-theoretic assumptions.

4 Generic Constructions from Identity-Based Extractable Hash Proofs

In this section, we present the generic constructions of IBE from (ABO) IB-EHPS. As a warm up, we first show the transformation from IB-EHPS to adaptive-identity CPA-secure IBE, then the transformation from ABO IB-EHPS to adaptive-identity CCA-secure IBE. Before going into details, we first give an intuitive explanation of the constructions from IB-EHPS to IBE with respect to the underlying relation. Suppose that the binary relation of an IB-EHPS is R_{pp} and (x, w) is a tuple that belongs to R_{pp}. The overall construction is: first encrypt (or commit to) a fresh DEM key (the corresponding witness is w) which is in turn used to encrypt the actual message, and then provide an identity-based extractable hash proof $y = \mathsf{H}_{mpk}(id, x)$ (which is also zero-knowledge) of the key. The ciphertext is of the form (x, y). In fact, such an approach was used implicitly in the PKE schemes based on computational assumptions and its connection to the Rackoff-Simon paradigm [27] was made explicit in [31]. Here we make its link to the underlying relation R clear. It is useful to note the distinguished feature in the construction from IB-EHPS to IBE that the value w (used to compute the session key) is uniquely determined by PP and the random coins used by SampR. This explains why IB-EHPS cannot encompass the IBE schemes whose session keys are related to the identity, e.g. Boneh-Franklin IBE [7].

4.1 IND-ID-CPA Secure IBE

Starting from an IB-EHPS ($\mathsf{SetupExt}$, $\mathsf{SetupHash}$, Pub, Priv, Ext, KeyGen, KeyGen^*) associating with a one-way relation instance $(X, W, \mathrm{PP}, \mathsf{R}_{\mathrm{pp}})$ and a hash family $\mathbf{H} = (\mathsf{H}, MPK, I, X, Y)$, we construct an IB-KEM as follows:

- $\mathsf{Setup}(\kappa)$: same as $\mathsf{SetupExt}(\mathrm{PP})$ in IB-EHPS.
- $\mathsf{KeyGen}(msk, id)$: same as $\mathsf{KeyGen}(msk, id)$ in IB-EHPS.
- $\mathsf{Encap}(id)$: samples $(x, w) \leftarrow \mathsf{SampR}(r)$, computes $y = \mathsf{Pub}(mpk, id, r)$, and returns a ciphertext $c = (x, y)$ and a DEM key $k = \mathsf{F}(w)$,
- $\mathsf{Decap}(sk, c)$: parses c as (x, y), and returns $\mathsf{F}(\mathsf{Ext}(sk, x, y))$.

The functionality of the above IB-KEM follows readily from the correctness of the extraction mode. For the security, we have the following theorem whose proof appears in the full version of this paper.

Theorem 4.1 *If* R_{pp} *is a one-way relation and the IB-EHPS is* (Q_e, δ)*-well-partition, then the above IB-KEM is IND-ID-CPA secure as long as* δ *is non-negligible.*

4.2 IND-ID-CCA Secure IBE

Starting from an ABO IB-EHPS ($\mathsf{SetupExt}$, $\mathsf{SetupABO}$, Pub, Priv, Verify, Verify^*, Ext, Ext^*, KeyGen, KeyGen^*) for a one-way relation instance $(X, W, \mathrm{PP}, \mathsf{R}_{\mathrm{pp}})$ and a hash family $\mathbf{H} = (\mathsf{H}, MPK, I, X, Y)$, we construct an IB-KEM as follows:

- $\mathsf{Setup}(\kappa)$: same as $\mathsf{SetupExt}(\mathrm{PP}, \mathrm{SP})$ in ABO IB-EHPS.
- $\mathsf{KeyGen}(msk, id)$: same as $\mathsf{KeyGen}(msk, id)$ in ABO IB-EHPS.
- $\mathsf{Encap}(id)$: samples $(x, w) \leftarrow \mathsf{SampR}(r)$, computes $y = \mathsf{Pub}(mpk, id, r)$, and returns a ciphertext $c = (x, y)$ and a associated DEM key $k = \mathsf{F}(w)$.
- $\mathsf{Decap}(sk, c)$: parses c as (x, y), and returns $\mathsf{F}(\mathsf{Ext}(sk, x, y))$.

The functionality of the above IB-KEM follows readily from the correctness of the extraction mode. For the security, we have the following theorem.

Theorem 4.2 *If* R_{pp} *is a one-way relation and the ABO IB-EHPS is* (Q_e, Q_d, δ)*-well-partition, then the above IB-KEM is IND-ID-CCA secure as long as* δ *is non-negligible.*

Proof. To establish the IND-ID-CCA security based on the one-wayness of relation R_{pp}, we proceed via a sequence of games. Let A be the event that $\mathcal{A}$ wins in Game CCA, and A_i be the event that $\mathcal{A}$ wins in Game i.

Game CCA. Given PP and SP, $\mathcal{CH}$ plays with $\mathcal{A}$ in the following game.
Setup: $\mathcal{CH}$ generates $(mpk, msk) \to \mathsf{SetupExt}(\mathrm{PP}, \mathrm{SP})$ and gives mpk to $\mathcal{A}$.
Phase 1 - Private key queries: When $\mathcal{A}$ submits a private key query $\langle id \rangle$, $\mathcal{CH}$ responds with $\mathsf{KeyGen}(msk, id)$.
Phase 1 - Decapsulation queries: When $\mathcal{A}$ submits a decapsulation query $\langle id, c = (x, y) \rangle$, $\mathcal{CH}$ extracts $sk = \mathsf{KeyGen}(msk, id)$ and responds with $\mathsf{Ext}(sk, x, y)$.
Challenge: When $\mathcal{A}$ submits a target identity id^* such that id^* did not appear in any private key query in Phase 1, $\mathcal{CH}$ samples $(x^*, w^*) \leftarrow \mathsf{SampR}(r^*)$ and computes $y^* = \mathsf{H}_{mpk}(id^*, x^*)$ via $\mathsf{Pub}(mpk, id^*, r^*)$, then sets $k_0^* = \mathsf{F}(w^*)$ and $k_1^* \xleftarrow{R} \{0,1\}^l$. $\mathcal{CH}$ picks $\beta \xleftarrow{R} \{0,1\}$ and returns (x^*, y^*, k_β^*) to $\mathcal{A}$ as the challenge.
Phase 2 - Private key queries: Same as in Phase 1 except that the query $\langle id^* \rangle$ is not allowed.
Phase 2 - Decapsulation queries: Same as in Phase 1 except that the query $\langle id^*, x^*, y^* \rangle$ is not allowed.
Guess: $\mathcal{A}$ outputs its guess β' for β and wins if $\beta' = \beta$.

$\mathcal{A}$'s view in Game CCA is identical to the standard IND-ID-CCA game, thus

$$\Pr[A] = 1/2 + \mathsf{Adv}_{\mathcal{A}}^{\mathrm{CCA}}(\kappa) \tag{5}$$

Game 0. Given PP and PP, $\mathcal{CH}$ plays with $\mathcal{A}$ in the following game.
Setup: Same as in Sub-Game 0 for ABO IB-EHPS.
Phase 1 - Private key queries: Same as in Sub-Game 0 for ABO IB-EHPS.
Phase 1 - Decapsulation queries: Same as in Sub-Game 0 for ABO IB-EHPS.
Challenge. Same as the Phase Middle in Sub-Game 0 for ABO IB-EHPS.
Phase 2 - Private key queries: Same as in Sub-Game 0 for ABO IB-EHPS.
Phase 2 - Decapsulation queries: Same as in Sub-Game 0 for ABO IB-EHPS.
Guess: $\mathcal{A}$ outputs its guess β' for β and wins if $\beta = \beta'$.

Observe that $\mathcal{A}$'s view in Game 0 is essentially the same as in Sub-Game 0. There are two differences between Game 0 and Game CCA: 1) in Game 0 the challenger samples (x^*, w^*) at the setup phase while in Game CCA the challenger samples (x^*, w^*) at the challenge phase. It is easy to see that this difference is invisible in $\mathcal{A}$'s view. 2) in Game 0 the challenger will return $\perp$ when encountering a decapsulation query with $x = x^*$ in Phase 1. We conclude that $\mathcal{A}$'s view in Game 0 is identical to Game CCA if the event that in Phase 1 $\mathcal{A}$ submits a decapsulation query with $x = x^*$ does not happen, whose probability is at most $Q_d/|X|$. Thus we have $|\Pr[A_0] - \Pr[A]| \leq Q_d/|X|$. Since $Q_d = \mathsf{poly}(\kappa)$, we have that $Q_d/|X| = \mathsf{negl}(\kappa)$ and hence $\Pr[A_0] \approx \Pr[A]$. We claim that $\mathsf{Adv}_{\mathcal{A}}^{\mathrm{CCA}} = \mathsf{negl}(\kappa)$ based on the one-wayness of R_{pp}. Suppose that there exists an algorithm $\mathcal{A}$ whose advantage against the CCA-security of IB-KEM is not negligible in κ, then we can construct an adversary $\mathcal{B}$ breaking the pseudo-randomness of F, which is sufficient to prove CCA-security under the one-wayness of R_{pp}.

Game 1. $\mathcal{B}$ receives a challenge instance (PP, x^*, k^*), where x^* is picked from the tuple $(x^*, w^*) \in \mathsf{R}_{\mathrm{pp}}$ generated by $\mathsf{SampR}(r^*)$ and k^* is either $\mathsf{F}(w^*)$ or randomly picked from $\{0,1\}^l$. $\mathcal{B}$ is asked to determine $k^* = \mathsf{F}(w^*)$ or $k^* \xleftarrow{R} \{0,1\}^l$. $\mathcal{B}$ plays with $\mathcal{A}$ in the following game.
Setup: $\mathcal{B}$ operates as $\mathcal{CH}$ does in Sub-Game 1 for ABO IB-EHPS except that $\mathcal{B}$ skips the sampling step.
Phase 1 - Private key queries: $\mathcal{B}$ operates as $\mathcal{CH}$ does in Sub-Game 1 for ABO IB-EHPS.
Phase 1 - Decapsulation queries: $\mathcal{B}$ operates as $\mathcal{CH}$ processes the decapsulation queries in Sub-Game 1 for ABO IB-EHPS.
Challenge: When $\mathcal{A}$ submits a target identity id^* on the condition that id^* did not appear in any private key query in Phase 1, $\mathcal{B}$ computes $y^* = \mathsf{H}_{mpk}(id^*, x^*)$ via $\mathsf{Priv}(msk^*, id^*, x^*)$, then instead of creating the challenge by explicitly generating a random bit β, it sends (x^*, y^*, k^*) to $\mathcal{A}$ as the challenge.
Phase 2 - Private key queries: $\mathcal{B}$ operates as $\mathcal{CH}$ does in Sub-Game 1 for ABO IB-EHPS.
Phase 2 - Decapsulation queries: $\mathcal{B}$ operates as $\mathcal{CH}$ processes the decapsulation queries in Sub-Game 1 for ABO IB-EHPS.
Guess: $\mathcal{A}$ outputs its guess β' for β and $\mathcal{B}$ forwards β' to its own challenger.

Observe that $\mathcal{A}$'s view in Game 1 is essentially the same as Sub-Game 1. Since the underlying ABO IB-EHPS is (Q_e, Q_d, δ)-well-partition, then we conclude that $\mathcal{B}$ can break the pseudo-randomness of F with advantage:

$$\mathsf{Adv}_{\mathcal{B}} = |(1-\delta)/2 + \delta \cdot \Pr[A_0] - 1/2| = \delta \cdot |\Pr[A_0] - 1/2| \approx \delta \cdot \mathsf{Adv}_{\mathcal{A}}^{\mathrm{CCA}}$$

If δ is non-negligible, then the above IB-KEM is IND-ID-CCA secure based on the one-wayness of R_{pp}. This proves the theorem. □

5 Instantiations of IB-EHPS

ABO IB-EHPS for the BDH Relation

We first run $\mathsf{SampS}(\kappa)$ to generate an instance $(X, W, \mathrm{PP}, \mathsf{R}_{\mathrm{pp}})$ of the BDH relation defined in Section 2.4, where $X = \mathbb{G}$, $W = \mathbb{G}_T$, $\mathrm{PP} = (g, g^a, g^b)$. $\mathbb{G}$ and $\mathbb{G}$ are two groups of prime order p and equipped with bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, and g is a random generator of $\mathbb{G}$. The random coins SP consumed by SampS consists of $(a, b) \in \mathbb{Z}_p^2$ and the randomness used to pick g. For the choice of $\mathbf{H} = (\mathsf{H}, MPK, I, X, Y)$, let $MPK = \mathbb{G}^{5+n}$ for some integer n, $I = \{0,1\}^n$, $Y = \mathbb{G}^2$. We write $\overline{u}$ for a n-length vector $(u_1, \ldots, u_n)$ hereafter. We also need a TCR hash function TCR from $\mathbb{G}$ to $\mathbb{Z}_p$. For $mpk = (g, g_1', g_1, g_2, u_0, \overline{u}) \in MPK$, we define:

$$\mathsf{H}_{mpk}(id, x) = (y_1, y_2) := ((g_1^t g_1')^r, F(id)^r)$$

Here $x = g^r$, $t = \mathsf{TCR}(x)$, and $F(id) = u_0 \prod_{i=1}^n u_i^{id_i}$ (id_i denotes the i-th bit of identity id) is known as Waters-hash.

Extraction Mode

- $\mathsf{SetupExt}(\mathrm{PP}, \mathrm{SP})$: sets $g = g$, $g_1 = g^a$, $g_2 = g^b$ (from PP), picks $g_1', u_0 \xleftarrow{R} \mathbb{G}$, $\overline{u} \xleftarrow{R} \mathbb{G}^n$, and returns $mpk = (g, g_1', g_1, g_2, u_0, \overline{u})$, $msk = a$ (from SP).
- $\mathsf{Pub}(mpk, id, r)$: returns $(y_1, y_2) = ((g_1^t g_1')^r, F(id)^r)$ where $t = \mathsf{TCR}(g^r)$.
- $\mathsf{KeyGen}(msk, id)$: picks $s \xleftarrow{R} \mathbb{Z}_p$, and returns $sk = (g_2^a F(id)^s, g^s)$.
- $\mathsf{Verify}(id, sk, x, y)$: parses y as (y_1, y_2), computes $t = \mathsf{TCR}(x)$, if $e(x, g_1^t g_1') = e(g, y_1)$ and $e(x, F(id)) = e(g, y_2)$ returns 1, else returns 0.
- $\mathsf{Ext}(sk, x, y)$: parses sk as (sk_1, sk_2) and y as (y_1, y_2), if $\mathsf{Verify}(id, sk, x, y) = 1$ then returns $e(x, sk_1)/e(y_2, sk_2)$, else returns $\perp$.

The correctness of extraction follows from the following simple calculation:

$$y = ((g_1^t g_1')^r, F(I)^r) = \mathsf{H}_{mpk}(I, u) \Longrightarrow e(x, g_2^a F(I)^s)/e(y_2, g^s) = e(g_1, g_2)^r$$

ABO Hashing Mode

- $\mathsf{SetupABO}(\mathrm{PP}, x^*)$: sets $g = g$, $g_1 = g^a$, $g_2 = g^b$ (from PP), picks $d \xleftarrow{R} \mathbb{Z}_p$, computes $t^* = \mathsf{TCR}(x^*)$, sets $g_1' = g_1^{-t^*} g^d$; sets $m = 2(Q_e + Q_d)$, and chooses $k \xleftarrow{R} [n+1]$; picks $\alpha' \xleftarrow{R} \mathbb{Z}_m$, $\overline{\alpha} \xleftarrow{R} \mathbb{Z}_m^n$, $\beta' \xleftarrow{R} \mathbb{Z}_p$, $\overline{\beta} \xleftarrow{R} \mathbb{Z}_p^n$, sets $u_0 = g_2^{p-km+\alpha'} g^{\beta'}$ and $u_i = g_2^{\alpha_i} g^{\beta_i}$ for $1 \le i \le n$; returns $mpk = (g, g_1', g_1, g_2, u_0, \overline{u})$, $msk^* = (t^*, d, \alpha', \overline{\alpha}, \beta', \overline{\beta})$. For ease of narration we define two functions, namely $J(id) = (p - mk) + \alpha' + \sum \alpha_i id_i$ and $K(id) = \beta' + \sum \beta_i id_i$. Hence $F(id)$ is essentially of the form $g_2^{J(id)} g^{K(id)}$. The structure of mpk implicitly splits the whole identity space I into I_1 and I_2. For an identity $id \in I$, if $J(id) \neq p$ it belongs to I_1, otherwise it belongs to I_2.

- $\mathsf{Priv}(msk^*, id, x)$: if $id \in I_2$ and $x = x^*$ returns $(y_1, y_2) = ((x^*)^d, (x^*)^{K(id)})$, else returns $\perp$.
- $\mathsf{Verify}^*(id, msk^*, x, y)$: same as Verify.
- $\mathsf{KeyGen}^*(msk^*, id)$: if $id \notin I_1$ returns $\perp$, else picks $s \xleftarrow{R} \mathbb{Z}_p$ and returns
$$sk = (sk_1, sk_2) = \left(g_1^{\frac{-K(id)}{J(id)}} F(id)^s, g_1^{\frac{-1}{J(id)}} g^s \right)$$
- $\mathsf{Ext}^*(msk^*, x, y)$: parses y as (y_1, y_2), if $\mathsf{Verify}^*(id, msk^*, x, y) = 1$ and $t \neq t^*$ then returns $e((y_1/x^d)^{1/(t-t^*)}, g_2)$ where $t = \mathsf{TCR}(x)$, else returns $\perp$.

The correctness of ABO hashing mode follows from the following two facts:

1. If $id \in I_2$ and $x = x^*$, we have $\mathsf{Priv}(msk^*, id, x^*) = ((x^*)^d, (x^*)^{K(id)}) = ((g^d)^{r^*}, (g^{K(id)})^{r^*}) = ((g_1^{t^*} g_1')^{r^*}, F(id)^{r^*}) = \mathsf{H}_{mpk}(id, x^*)$.
2. If $x \neq x^*$, then $((g_1^t g_1')^r, F(id)^r) = \mathsf{H}_{mpk}(id, x) \Longrightarrow e((y_1/x^d)^{1/(t-t^*)}, g_2) = e(g_1, g_2)^r)$, where $t = \mathsf{TCR}(x)$. The property of TCR ensures that $t = t^*$ holds with overwhelming probability when $x = x^*$.

The indistinguishability is established from the following two facts:

1. The distribution of mpk in both modes are identical.
2. For any mpk and any identity $id \in I_1$, the output of $\mathsf{KeyGen}(msk, id)$ and $\mathsf{KeyGen}^*(msk^*, id)$ are statistically indistinguishable. To see this, let $\tilde{s} = s - a/J(id)$, we have
$$sk_1 = g_1^{\frac{-K(id)}{J(id)}} F(id)^s = g_1^{\frac{-K(id)}{J(id)}} (g_2^{J(id)} g^{K(id)})^s = g_2^a F(id)^{s - \frac{a}{J(id)}} = g_2^a F(id)^{\widetilde{s}}$$
$$sk_2 = g_1^{\frac{-1}{J(id)}} g^s = g^{s - \frac{a}{J(id)}} = g^{\widetilde{s}}$$
Since s is uniform in $\mathbb{Z}_p$, then $\tilde{s}$ is also uniform in $\mathbb{Z}_p$. Thereby the distribution of $\mathsf{KeyGen}(msk, id)$ and $\mathsf{KeyGen}^*(msk^*, id)$ are identical.

Follow the same analysis in [23], the above IB-EHPS is (Q_e, Q_d, δ)-well-partition, where $\delta \geq \frac{1}{8(n+1)(Q_e+Q_d)}$. Applying the transformation in Section 4.2 to this ABO IB-EHPS, we obtain an IB-KEM (see Fig. 1), which can be viewed as a variant of the IB-KEM in [23]. Combining theorem 4.2, we conclude that this IB-KEM is IND-ID-CCA secure based the CBDH assumption.

ABO IB-EHPS for the mBDH Relation

Based on the modified bilinear Diffie-Hellman relation $\mathsf{R}_{\mathsf{pp}}^{\mathsf{mbdh}}$, we can create an ABO IB-EHPS whose Ext and Ext^* algorithms implement the “implicitly rejection” idea. Applying the transformation from Section 4.2 to the ABO IB-EHPS, we obtain a CCA-secure IB-KEM based on the mBDH assumption (see Fig. 2), which is a variant of the IB-KEM in [24].

```
Setup(κ):
    $g, g_1', g_2, u_0 \xleftarrow{R} \mathbb{G}, \overline{u} \xleftarrow{R} \mathbb{G}^n$; $a \xleftarrow{R} \mathbb{Z}_p$
    $F(id) = u_0 \prod_{i=1}^{n} u_i^{id_i}$
    $mpk = (g, g_1 = g^a, g_1', g_2, u_0, \overline{u})$; $msk = a$
    return $(mpk, msk)$

Extract($msk, I$)
    $s \xleftarrow{R} \mathbb{Z}_p$
    $sk = (g_2^a F(id)^s, g^s)$
    return $sk$

Encap($id$)
    $r \xleftarrow{R} \mathbb{Z}_p$, $x \leftarrow g^r$
    $t \leftarrow \mathsf{TCR}(x)$
    $y_1 = (g_1^t g_1')^r$, $y_2 = F(id)^r$
    $k \leftarrow \mathsf{GL}(e(g_1, g_2)^r)$
    return $c = (x, y_1, y_2)$

Decap($sk, c$)
    parse $sk$ as $(sk_1, sk_2)$, $c$ as $(x, y_1, y_2)$
    $t = \mathsf{TCR}(x)$
    If $e(x, g_1^t g_1') \neq e(g, y_1)$ or
        $e(x, F(id)) \neq e(g, y_2)$, then return $\perp$
    else return $\mathsf{GL}(e(x, sk_1)/e(y_2, sk_2))$
```

Fig. 1. An IND-ID-CCA secure IB-KEM based on BDH (variant of [23])

```
Setup(κ):
    $g, g_2, u_0 \xleftarrow{R} \mathbb{G}, \overline{u} \xleftarrow{R} \mathbb{G}^n$; $a \xleftarrow{R} \mathbb{Z}_p$
    $F(id) = u_0 \prod_{i=1}^{n} u_i^{id_i}$
    $mpk = (g, g_1 = g^a, g_2, u_0, \overline{u})$; $msk = a$
    return $(mpk, msk)$

Extract($msk, I$)
    $s \xleftarrow{R} \mathbb{Z}_p$
    $sk = (g_2^a F(id)^s, g^{-s}, g_2^s)$
    return $sk$

Encap($id$)
    $r \xleftarrow{R} \mathbb{Z}_p$, $x \leftarrow g^r$, $t \leftarrow \mathsf{TCR}(x)$
    $y = (F(id) g_2^t)^r$, $k \leftarrow \mathsf{GL}(e(g_1, g_2)^r)$
    return $c = (x, y)$

Decap($sk, c$)
    parse $sk$ as $(sk_1, sk_2, sk_3)$, $c$ as $(x, y)$
    $t = \mathsf{TCR}(x)$
    return $\mathsf{GL}(e(x, sk_1 \cdot sk_3^t) \cdot e(y, sk_2))$
```

Fig. 2. An IND-ID-CCA secure IB-KEM based on mBDH (variant of [24])

6 Extension

We also put forward the notion of dual ABO IB-EHPS, which can be viewed as a special case of ABO IB-EHPS whose I_2 contains a single point id^*. The term "dual ABO" reflects that the algorithm Priv returns $\mathsf{H}_{mpk}(id, x)$ only on the point that $id = id^*$ and $x = x^*$. The dual ABO IB-EHPS turns out to be a useful paradigm for constructing selective-identity CCA-secure IB-KEM. In particular, the instantiation of dual ABO IB-EHPS from the BDH relation serves as a clarification of all the known selective-identity CCA-secure IB-KEMs [11,14,19] based on the CBDH assumption. Due to space limit, we include this part in the full version of this paper.

Acknowledgment. The authors greatly thank the anonymous ACNS 2012 reviewers for many helpful and insightful comments. This work is supported by the Foundation of SKLOIS of Chinese Academy of Sciences, the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDA06010701, the National 973 Program of China under Grant No. 2011CB302400, and the National Natural Science Foundation of China under Grant No. 60970152. The second and the fourth authors are partially supported by the National Natural Science Foundation of China under Grant Nos. 61033014, 60970110, 60972034 and Asia 3 Foresight Program under Grant No. 61161140320.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)
2. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: ACM Conference on Computers and Communication Security, pp. 62–73 (1995)
3. Bentahar, K., Farshim, P., Malone-Lee, J., Smart, N.P.: Generic constructions of identity-based and certificateless kems. Journal of Cryptology 21(2), 178–199 (2008)
4. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
5. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM J. Comput. 36(5), 1301–1328 (2007)
6. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
7. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. SIAM Journal on Computation 32, 586–615 (2003)
8. Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
9. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai Trees, or How to Delegate a Lattice Basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010)
10. Cash, D., Kiltz, E., Shoup, V.: The Twin Diffie-Hellman Problem and Applications. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Heidelberg (2008)
11. Chen, Y., Chen, L., Zhang, Z.: CCA secure IB-KEM from the computational bilinear diffie-hellman assumption in the standard model (2011), `http://eprint.iacr.org/2011/593`
12. Cocks, C.: An Identity Based Encryption Scheme Based on Quadratic Residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001)
13. 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)
14. Galindo, D.: Chosen-Ciphertext Secure Identity-Based Encryption from Computational Bilinear Diffie-Hellman. In: Joye, M., Miyaji, A., Otsuka, A. (eds.) Pairing 2010. LNCS, vol. 6487, pp. 367–376. Springer, Heidelberg (2010)
15. Gentry, C.: Practical Identity-Based Encryption Without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)

16. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC, pp. 197–206. ACM (2008)
17. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing, STOC, pp. 25–32. ACM (1989)
18. Hanaoka, G., Kurosawa, K.: Efficient Chosen Ciphertext Secure Public Key Encryption under the Computational Diffie-Hellman Assumption. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 308–325. Springer, Heidelberg (2008)
19. Haralambiev, K., Jager, T., Kiltz, E., Shoup, V.: Simple and Efficient Public-Key Encryption from Computational Diffie-Hellman in the Standard Model. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 1–18. Springer, Heidelberg (2010)
20. Hofheinz, D., Kiltz, E.: Practical Chosen Ciphertext Secure Encryption from Factoring. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 313–332. Springer, Heidelberg (2009)
21. Kiltz, E.: Chosen-Ciphertext Security from Tag-Based Encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
22. Kiltz, E.: Chosen-Ciphertext Secure Key-Encapsulation Based on Gap Hashed Diffie-Hellman. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 282–297. Springer, Heidelberg (2007)
23. Kiltz, E., Galindo, D.: Direct Chosen-Ciphertext Secure Identity-Based Key Encapsulation Without Random Oracles. In: Batten, L.M., Safavi-Naini, R. (eds.) ACISP 2006. LNCS, vol. 4058, pp. 336–347. Springer, Heidelberg (2006)
24. Kiltz, E., Vahlis, Y.: CCA2 Secure IBE: Standard Model Efficiency through Authenticated Symmetric Encryption. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 221–238. Springer, Heidelberg (2008)
25. Micali, S., Rackoff, C., Sloan, B.: The notion of security for probabilistic cryptosystems. SIAM J. Comput. 17(2), 412–426 (1988)
26. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC 2008, pp. 187–196 (2008)
27. Rackoff, C., Simon, D.R.: Non-interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
28. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
29. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
30. Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)
31. 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)