Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions

Sherman S.M. Chow
New York University
schow@cs.nyu.edu

Yevgeniy Dodis
New York University
dodis@cs.nyu.edu

Yannis Rouselakis
The University of Texas at Austin
jrous@cs.utexas.edu

Brent Waters
The University of Texas at Austin
bwaters@cs.utexas.edu

ABSTRACT

We design the first *Leakage-Resilient Identity-Based Encryption* (LR-IBE) systems from static assumptions in the standard model. We derive these schemes by applying a hash proof technique from Alwen *et al.*(Eurocrypt '10) to variants of the existing IBE schemes of Boneh-Boyen, Waters, and Lewko-Waters. As a result, we achieve leakage-resilience under the respective static assumptions of the original systems in the standard model, while also preserving the efficiency of the original schemes. Moreover, our results extend to the Bounded Retrieval Model (BRM), yielding the first regular and identity-based BRM encryption schemes from static assumptions in the standard model.

The first LR-IBE system, based on Boneh-Boyen IBE, is only selectively secure under the simple *Decisional Bilinear Diffie-Hellman* assumption (DBDH), and serves as a stepping stone to our second fully secure construction. This construction is based on Waters IBE, and also relies on the simple DBDH. Finally, the third system is based on Lewko-Waters IBE, and achieves full security with shorter public parameters, but is based on three static assumptions related to composite order bilinear groups.

Categories and Subject Descriptors

E.3 [**Data Encryption**]: Public key cryptosystems

General Terms

Security, Algorithms, Design

Keywords

identity based encryption, hash proof system, leakage resilience, bounded retrieval model, dual system encryption

CCS'10, October 4–8, 2010, Chicago, Illinois, USA.

1. INTRODUCTION

Traditionally in cryptography, we assume that the secret keys are completely hidden from the potential attackers. However, several works [26, 27, 23] showed that this premise is not necessarily true in real systems. Many attacks such as timing attacks, power dissipation, cold-boot attacks, can extract some bits of information from the secret keys or the state of the encrypting system, compromising security. In response to this, there has been a surge of interest in creating "leakage-resilient" cryptographic schemes. While there has already been many models of "leakage-resilience", which we survey later, the common goal is to design cryptographic systems resilient to large amounts of leakage with (1) comparable efficiency to previously known systems, and (2) security based on simple assumptions in the standard model.

In this work we concentrate on the model of *memory attacks*, or the *relative-leakage* model, introduced by Akavia *et al.* [1] in response to the cold-boot attack [23]. In this model the attacker can learn *any efficiently computable function of the system's secret key*, subject only to the constraint that the total amount of information learned is bounded by ℓ bits. Here ℓ is some "leakage parameter" of the system, typically chosen relative to the security parameter of the system, with the goal of making ℓ close to the length of the secret key.

Despite its recent introduction, the relative leakage model has already attracted considerable attention [1, 30, 3, 25, 2, 11], due to its elegance and generality. In particular, one natural application where protection against memory attacks is very relevant [1, 2] is the setting of *identity-based encryption* (IBE) [34]. An IBE system gives the ability to different parties to encrypt messages knowing only the identity of the receiver. The identities are used in a way similar to public encryption keys, so we avoid the problem of public-key distribution. In the context of leakage-resilience, it is natural to ask the question if one can design *leakage-resilient* IBE systems satisfying the desirable properties (1)-(2) above.

As it turns out, existing leakage-resilient IBE schemes [1, 2] are based on known regular IBE schemes [21, 20, 6], also having very similar efficiency, and hence satisfying our goal (1). Unfortunately, the security of these schemes is either analyzed in the random oracle model or is based on "non-static" assumption in the standard model, therefore violating property (2). Indeed, the lattice-based leakage-resilient schemes [1, 2], derived from the IBE of Gentry *et al.* [21], are either analyzed in the random oracle model (and the standard learning with errors assumption), or assume a highly non-standard interactive assumption. The same is true for the quadratic-residuosity based scheme [2] derived from the IBE of Boneh *et al.* [6]. The only leakage-resilient IBE in the standard model (and without resorting to interactive assumptions) is the scheme [2] derived from the IBE of Gentry [20]. However, it is only proven secure under a complex "q-type" assumption , where the size grows linearly with the number q of attacker's queries. Given that there exist IBE systems secure under simple static assumptions in the standard model [4, 35, 28], it is a natural question to ask whether we can give *leakage-resilient* versions of these systems. This was posed as an open question by Alwen *et al.* [2].

Our Results. We resolve this question in the affirmative, and derive three leakage-resilient IBE schemes. Our first system is based on Boneh-Boyen IBE [4]. This is only selectively-secure but it serves as a simpler version of the fully secure second system based on Waters IBE [35]. Like the original schemes, we prove that both variants are leakage-resilient under the decisional bilinear Diffie-Hellman assumption (DBDH). This is a well-studied static assumption used many times in various constructions. However, the second system has large public parameters. In order to overcome this obstacle we present a third system based on Lewko-Waters IBE [28], which is secure under three static assumptions related to bilinear composite order groups. These assumptions can be shown to hold in the generic group model if factoring is hard [28]. Efficiency results of the new systems compared to the old ones are shown in Table 1.

Our Technique. First, we use the hash proof system technique of Alwen *et al.* [2], who showed that one can build a leakage-resilient IBE (LR-IBE) from what they call an *identity-based hash proof system* (IB-HPS). *Very informally*, one can think of an IB-HPS is a special kind of IBE where, for each identity id, there are many valid secret keys $\mathsf{sk}_{\mathsf{id}}$ and also two kinds of ciphertexts: valid and invalid. The valid ciphertext C decrypts the same with any secret key $\mathsf{sk}_{\mathsf{id}}$, while an invalid ciphertext C' decrypts to a random value R' under a random possible secret key $\mathsf{sk}_{\mathsf{id}}$. Moreover, a "random" valid ciphertext C for id is indistinguishable from a "random" invalid ciphertext C' for id, even given the *full* secret key $\mathsf{sk}_{\mathsf{id}}$. The two properties immediately imply that an IB-HPS is also a secure IBE when no leakage is allowed. Moreover, even given ℓ bits of leakage about $\mathsf{sk}_{\mathsf{id}}$, decrypting a valid ciphertext C produces a value R which is indistinguishable from a value R' having $(|R| - \ell)$ bits of entropy. Thus, one can get the desired LR-IBE by combining an IB-HPS with a well studied primitive called a *randomness extractor* (see [31, 13]), which will extract from R a one-time pad of length (almost) $(|R| - \ell)$ to mask the actual message.

The advantage of dealing with IB-HPS rather than LR-IBE is that its security definition is much simpler to state, as it does not deal with leakage, and it also abstracts away the use of the randomness extractor in the final LR-IBE construction. However, IB-HPS are harder to construct than regular IBEs, as one needs to come up with an invalid ciphertext generation algorithm. Luckily, Alwen *et al.* [2] observed that this task is essentially immediate for some existing IBEs [20, 6, 21]. Coincidentally, these are precisely the IBEs analyzed in the random oracle model or based on non-static assumptions. On the other hand, it is not clear how to define invalid ciphertext generation for some other IBEs in the standard model [4, 35, 28]. One of the key hurdles comes from the fact that in these IBEs the owner of the secret key could easily re-randomize his key arbitrarily, without any other secret information. Unfortunately, this property, which was important to argue the security of these schemes, makes it impossible to turn these schemes into hash proof systems. Namely, to define invalid ciphertexts which are indistinguishable from valid ones, but decrypt to random values under random $\mathsf{sk}_{\mathsf{id}}$. Indeed, given a challenge ciphertext C and a secret key $\mathsf{sk}_{\mathsf{id}}$, the attacker can produce a random key $\mathsf{sk}'_{\mathsf{id}}$ and decrypt C twice with $\mathsf{sk}_{\mathsf{id}}$ and $\mathsf{sk}'_{\mathsf{id}}$. If the results are the same, C is valid; otherwise, C is invalid.

In this work we develop a new technique which allows us to circumvent this problem, and eventually build the desired IB-HPS's with almost the same complexity as the original IBEs. The idea is to add another degree of randomness to our identity-based secret keys, called the "tag" t, coupled with some master secret key terms. This is done in a way that the secret-key holder can now only re-randomize his key along the original degree of freedom (which is needed for the original proof), but cannot re-randomize the key along the new "tag-dimension" anymore. This will let us define invalid ciphertexts which decrypt to random values when the tag t is random, and yet decrypt to the same value when the tag t is kept the same, but the key is re-randomized along the original degree of freedom. This high-level technique is the main technical contribution of our work, and will be explained in more detail for the specific schemes.

Implications to Bounded Retrieval Model. As another advantage of building an IB-HPS rather than an LR-IBE, Alwen *et al.* [2] also showed how to use any IB-HPS to construct public-key and identity-based encryption schemes in the so called *Bounded Retrieval Model* (BRM).[1] Informally, a BRM scheme strengthens the relative leakage model by allowing one to arbitrarily increase the leakage parameter ℓ by *only* increasing the secret key of the system, but without significantly increasing the size of public parameters and the encryption/decryption times (i.e., those remain essentially independent on ℓ and only depend on the security parameter). We refer the reader to [2] for more discussion, but point out that our new constructions of identity-based hash proof systems from static assumption immediately imply the corresponding constructions of BRM-secure public-key and identity-based encryption schemes from static assumptions. No such constructions were known prior to our work.

Related Work. IBE was first proposed in [34] and the first construction, secure in the random oracle model, was given in [5]. By utilizing a weaker notion of security, known as selective security, many IBE systems were built in the standard model [4, 8]. The first fully secure and efficient IBE system based on a simple assumption was given in [35].

Leakage-resilient systems on the other hand present more diversity and different models have been proposed. Some early models severely restricted the classes of allowed leakages available to the attacker [33, 7, 24, 19]. More recently, Micali and Reyzin [29] proposed a leakage model called "only computation leaks information", where unbounded amount of leakage is allowed but only from parts of memory that are accessed, and several schemes in these model were recently proposed [17, 32, 18, 14]. A different model of leak-

[1]Moreover, this construction does not generally work with an LR-IBE in place of IB-HPS [2].

IBE System	Enc.Time	Dec.Time	Ciphertext Size	Parameters Size	Problem	Security	Leakage
Boneh-Boyen [4]	$4 \cdot E$	$2 \cdot P$	$1 \cdot R_T + 2 \cdot R$	$1 \cdot R_T + 3 \cdot R$	DBDH	Selective	0
L-R BB (Sec. 3)	$5 \cdot E$	$2 \cdot P$	$1 \cdot R_T + 2 \cdot R + 1 \cdot X$	$2 \cdot R_T + 3 \cdot R$	DBDH	Selective	1/3
Waters [35]	$3 \cdot E$	$2 \cdot P$	$1 \cdot R_T + 2 \cdot R$	$1 \cdot R_T + (B+2) \cdot R$	DBDH	Full	0
L-R W (Sec. 4)	$4 \cdot E$	$2 \cdot P$	$1 \cdot R_T + 2 \cdot R + 1 \cdot X$	$2 \cdot R_T + (B+2) \cdot R$	DBDH	Full	1/3
Lewko-Waters [28]	$4 \cdot E$	$2 \cdot P$	$1 \cdot R_T + 2 \cdot R$	$1 \cdot R_T + 3 \cdot R$	1,2,3	Full	0
L-R LW (Sec. 5)	$5 \cdot E$	$2 \cdot P$	$1 \cdot R_T + 2 \cdot R + 1 \cdot X$	$2 \cdot R_T + 3 \cdot R$	1,2,3	Full	1/9

Table 1: Efficiency results for existing systems and our constructions

L-R denotes the leakage-resilient version of each system in this paper. For encryption and decryption times we count only the dominant operations, which are the exponentiations in $\mathbb{G}$ and $\mathbb{G}_{\mathbb{T}}$ (both denoted as E), and the pairings (denoted as P), respectively. For sizes we denote by R_T, R the number of bits for the representation of elements of $\mathbb{G}_T$ and $\mathbb{G}$, respectively. X is the size of plaintext messages (typically symmetric encryption keys) plus the size of the extractor's seed. B is the number of bits of each identity. 1,2,3 are the assumptions on composite order groups given in Section 2.5, the operation times and the parameter sizes in these systems are thus relatively larger. Leakage refers to the formally shown tolerable relative leakage.

age, which is the model used in this work and which captures the cold-boot memory attacks, allows the attacker to call an arbitrary leakage function on the secret key. Naturally, the overall amount of leakage has to be bounded in this case, since otherwise an attacker can get the entire secret key. Two different models have been proposed: the relative leakage model [1, 30, 3, 25, 2, 12, 11], where the leakage is a portion of the secret key and depends on the security parameter, and the bounded retrieval model [15, 10, 16, 2, 3], which allows for arbitrary large leakage and increasing sizes of secret keys, but with constant cost of encryption and decryption unrelated to the amount of tolerable leakage.

2. PRELIMINARIES

Notations. For $n \in \mathbb{N}$, 1^n denotes a string of ones. We write $[n]$ for $\{1, 2, \cdots, n\}$. By $\mathsf{negl}(n)$ we denote a negligible function of n. We use $s \xleftarrow{\$} \mathcal{S}$ to denote that s is picked uniformly at random from the set $\mathcal{S}$. We write PPT for probabilistic polynomial time. Finally, the statistical distance between two random variables X, Y over a finite domain Ω is defined as $\mathbf{SD}(X, Y) = \frac{1}{2}\sum_{\omega\in\Omega} |\Pr[X = \omega] - \Pr[Y = \omega]|$.

2.1 Identity-Based Encryption

An identity-based encryption scheme [34] consists of four PPT algorithms ($\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Encrypt}, \mathtt{Decrypt}$).

IBE

$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathtt{Setup}(1^\lambda)$ **:** The setup algorithm takes a security parameter λ (in unary) and produces the *master public key* mpk (which defines an *identity set* $\mathcal{ID}$ and a *message space* $\mathcal{M}$) and the *master secret key* msk. All other algorithms $\mathtt{KeyGen}, \mathtt{Encrypt}, \mathtt{Decrypt}$ implicitly include mpk as an input.

$\mathsf{sk}_{\mathsf{id}} \leftarrow \mathtt{KeyGen}(\mathsf{id}, \mathsf{msk})$ **:** For any identity $\mathsf{id} \in \mathcal{ID}$, the user secret key generation algorithm uses the master secret key msk to sample an identity secret key $\mathsf{sk}_{\mathsf{id}}$.

$\mathsf{C} \leftarrow \mathtt{Encrypt}(\mathsf{id}, M)$ **:** The encryption algorithm takes an identity id, and a message M to be encrypted, outputs an encryption C of the message M for identity id

$M \leftarrow \mathtt{Decrypt}(\mathsf{C}, \mathsf{sk}_{\mathsf{id}})$ **:** The decryption algorithm takes a ciphertext C and a secret key of identity id, and outputs the message M (provided the ciphertext was a correct encryption for identity id).

We require that an IBE satisfies the following properties.

I. Correctness of Decryption. For any $(\mathsf{mpk}, \mathsf{msk})$ produced by $\mathtt{Setup}(1^\lambda)$, any $\mathsf{id} \in \mathcal{ID}$, any $M \in \mathcal{M}$, we have

$$\Pr\left[M' \neq M \;\middle|\; \begin{array}{c} \mathsf{sk}_{\mathsf{id}} \leftarrow \mathtt{KeyGen}(\mathsf{id}, \mathsf{msk}) \\ \mathsf{C} \leftarrow \mathtt{Encrypt}(\mathsf{id}, M) \\ M' \leftarrow \mathtt{Decrypt}(\mathsf{C}, \mathsf{sk}_{\mathsf{id}}) \end{array} \right] \leq \mathsf{negl}(\lambda).$$

II. Semantic Security with Leakage. We follow the natural definition from [1, 2], which roughly states that an IBE is *ℓ-leakage-resilient* if it remains secure (in the standard sense defined in [5]) despite the fact that the attacker can learn up to ℓ bits of arbitrary information about the secret key of the identity id^* he wants to attack. We notice that this definition also has the restriction that only one secret key can be produced/leaked for each identity id (even those different from id^*). For simplicity, we also follow this model, but remark that our IBE systems can be proven secure in a slightly stronger model, where secret key queries from multiple secret keys for identities other than id^* is allowed.

The resulting notion, called semantic security with leakage, is parameterized by the game $\mathsf{IBE\text{-}SSL}(\lambda, \ell)$, where λ is a security parameter and $\ell = \ell(\lambda)$ is a leakage parameter.

$\mathsf{IBE\text{-}SSL}(\lambda, \ell)$

Setup : The challenger $\mathcal{C}$ computes $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathtt{Setup}(1^\lambda)$ and gives mpk to the adversary $\mathcal{A}$.

Test Stage 1 : $\mathcal{A}$ can adaptively ask $\mathcal{C}$ for the following:

- SecretKey : On input $\mathsf{id} \in \mathcal{ID}$, $\mathcal{C}$ replies with $\mathsf{sk}_{\mathsf{id}}$.
- Leak : On input $\mathsf{id} \in \mathcal{ID}$, a PPT function $f : \mathcal{ID} \rightarrow \{0,1\}$, $\mathcal{C}$ replies with $f(\mathsf{sk}_{\mathsf{id}})$.

Challenge Stage : $\mathcal{A}$ selects two messages $M_0, M_1 \in \mathcal{M}$ and a challenge identity $\mathsf{id}^* \in \mathcal{ID}$ which *never appeared* in a secret-key query and appeared in *at most* ℓ leakage queries. $\mathcal{C}$ chooses $b \leftarrow \{0,1\}$ uniformly at random and gives $\mathsf{C} \leftarrow \mathtt{Encrypt}(\mathsf{id}^*, M_b)$ to the adversary $\mathcal{A}$.

Test Stage 2 : $\mathcal{A}$ gets to make *secret-key queries* SecretKey for arbitrary $\mathsf{id} \neq \mathsf{id}^*$. $\mathcal{C}$ replies with $\mathsf{sk}_{\mathsf{id}}$.

Output : $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ and *wins* if $b' = b$.

Note: For secret-key or leakage queries, $\mathcal{C}$ computes $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathtt{KeyGen}(\mathsf{id}, \mathsf{msk})$ the first time that id is queried and responds to all future queries on the same id with the same $\mathsf{sk}_{\mathsf{id}}$.

The *advantage* of an adversary $\mathcal{A}$ in the *semantic security game with leakage* ℓ is $\mathsf{Adv}^{\mathsf{IBE\text{-}SSL}}_{\mathsf{IBE},\mathcal{A}}(\lambda,\ell) \stackrel{\text{def}}{=} \left|\Pr[\mathcal{A} \text{ wins}] - \frac{1}{2}\right|$.

Definition 2.1 (Leakage-Resilient IBE). *An IBE scheme is* ℓ-leakage-resilient, *if the advantage of any PPT adversary* $\mathcal{A}$ *in the semantic security game with leakage* ℓ, *is* $\mathsf{Adv}^{\mathsf{IBE\text{-}SSL}}_{\mathsf{IBE},\mathcal{A}}(\lambda,\ell) \le \mathsf{negl}(\lambda)$. *We define the* relative leakage *of the scheme to be* $\hat{\alpha} \stackrel{def}{=} \ell/\hat{\mu}$, *where* $\hat{\mu}$ *is the number of bits needed to efficiently store identity secret keys* $\mathsf{sk}_{\mathsf{id}}$.

The notion is indeed very similar to the traditional notion of semantic security for IBE [22, 5], except for the introduction of the leakage queries $\mathsf{Leak}(\mathsf{id}, f_i)$, where $f_i : \mathcal{SK} \to \{0,1\}$ is any efficiently computable function. Without loss of generality, we restrict the output of each such f_i to a single bit, but clearly the definition implies the ones where the attacker can get multiple bits per leakage query, as long as their total length for each identity is most ℓ bits. As remarked, we assume that each leaked secret key $\mathsf{sk}_{\mathsf{id}}$ has to be the same in all subsequent calls to $\mathsf{Leak}(\mathsf{id},\cdot)$.

Selective Security. If we modify the above security game so that the adversary gives the challenge identity id^* to the challenger before the setup, we get the "selective security" game IBE-sSSL, with the corresponding advantage $\mathsf{Adv}^{\mathsf{IBE\text{-}sSSL}}_{\mathsf{IBE},\mathcal{A}}(\lambda,\ell) = \Pr[\mathcal{A} \text{ wins}] - \frac{1}{2}$. This, in turn, yields the notion of *selectively* ℓ*-leakage resilient secure IBE.*

2.2 Identity-Based Hash Proof System

An *Identity-Based Hash Proof System* (IB-HPS) consists of five PPT algorithms: $(\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Encap}, \mathtt{Encap}^*, \mathtt{Decap})$. The algorithms have the following syntax.

IB-HPS

$(\mathsf{mpk},\mathsf{msk}) \leftarrow \mathtt{Setup}(1^\lambda)$ **:** The setup algorithm takes as input a security parameter λ and produces the *master public key* mpk and the *master secret key* msk. The master public key defines an *identity set* $\mathcal{ID}$, and an *encapsulated-key set* $\mathcal{K}$. All other algorithms $\mathtt{KeyGen}, \mathtt{Encap}, \mathtt{Decap}, \mathtt{Encap}^*$ implicitly include mpk as an input.

$\mathsf{sk}_{\mathsf{id}} \leftarrow \mathtt{KeyGen}(\mathsf{id},\mathsf{msk})$ **:** For any identity $\mathsf{id} \in \mathcal{ID}$, the $\mathtt{KeyGen}$ algorithm uses the master secret key msk to sample an identity secret key $\mathsf{sk}_{\mathsf{id}}$.

$(\mathsf{C},k) \leftarrow \mathtt{Encap}(\mathsf{id})$ **:** The *valid* encapsulation algorithm creates pairs (C,k) where C is a valid ciphertext, and $k \in \mathcal{K}$ is the encapsulated-key.

$\mathsf{C} \leftarrow \mathtt{Encap}^*(\mathsf{id})$ **:** The alternative *invalid* encapsulation algorithm samples an invalid ciphertext C for a given id.

$k \leftarrow \mathtt{Decap}(\mathsf{C},\mathsf{sk}_{\mathsf{id}})$ **:** The decapsulation algorithm is deterministic, takes a ciphertext C and an identity secret key $\mathsf{sk}_{\mathsf{id}}$, and outputs the encapsulated key k.

We require that an IB-HPS satisfies the following properties.

I. Correctness of Decapsulation. For any values of $\mathsf{mpk},\mathsf{msk}$ produced by $\mathtt{Setup}(1^\lambda)$, any $\mathsf{id} \in \mathcal{ID}$, we have

$$\Pr\left[k \neq k' \;\middle|\; \begin{array}{c} \mathsf{sk}_{\mathsf{id}} \leftarrow \mathtt{KeyGen}(\mathsf{id},\mathsf{msk}) \\ (\mathsf{C},k) \leftarrow \mathtt{Encap}(\mathsf{id}) \\ k' = \mathtt{Decap}(\mathsf{C},\mathsf{sk}_{\mathsf{id}}) \end{array} \right] \le \mathsf{negl}(\lambda).$$

II. Valid/Invalid Ciphertext Indistinguishability. The valid ciphertexts generated by $\mathtt{Encap}$ and the invalid ciphertexts generated by $\mathtt{Encap}^*$ should be indistinguishable *even given the identity secret key.* This property is captured in the following distinguishability game.

VI-IND(λ)

Setup : The challenger $\mathcal{C}$ computes $(\mathsf{mpk},\mathsf{msk}) \leftarrow \mathtt{Setup}(1^\lambda)$ and gives mpk to the adversary $\mathcal{A}$.

Test Stage 1 : The adversary $\mathcal{A}$ adaptively queries the challenger $\mathcal{C}$ with $\mathsf{id} \in \mathcal{ID}$ and $\mathcal{C}$ responds with $\mathsf{sk}_{\mathsf{id}}$.

Challenge Stage : $\mathcal{A}$ selects an *arbitrary* challenge identity $\mathsf{id}^* \in \mathcal{ID}$, and $\mathcal{C}$ chooses $b \leftarrow \{0,1\}$.
If $b = 0$, $\mathcal{C}$ computes $(\mathsf{C},k) \leftarrow \mathtt{Encap}(\mathsf{id}^*)$.
If $b = 1$, $\mathcal{C}$ computes $\mathsf{C} \leftarrow \mathtt{Encap}^*(\mathsf{id}^*)$.
$\mathcal{C}$ gives C to the adversary $\mathcal{A}$.

Test Stage 2 : $\mathcal{A}$ adaptively queries the challenger with $\mathsf{id} \in \mathcal{ID}$ and $\mathcal{C}$ responds with $\mathsf{sk}_{\mathsf{id}}$.

Output : $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ which is the output of the game. We say that $\mathcal{A}$ *wins* the game if $b' = b$.

Note: In both test stages, $\mathcal{C}$ computes $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathtt{KeyGen}(\mathsf{id},\mathsf{msk})$ the first time that id is queried and responds to all future queries on the same id with the same $\mathsf{sk}_{\mathsf{id}}$.

Note that, during the challenge stage, the adversary can choose *any* identity id^*, and possibly even one for which it has seen the secret key $\mathsf{sk}_{\mathsf{id}^*}$ in Test Stage 1 (or the adversary can simply get $\mathsf{sk}_{\mathsf{id}^*}$ in Test Stage 2). Without loss of generality, we assume that the adversary always asks for $\mathsf{sk}_{\mathsf{id}^*}$ right before the challenge stage. We define the advantage of $\mathcal{A}$ in distinguishing valid/invalid ciphertexts to be $\mathsf{Adv}^{\mathsf{VI\text{-}IND}}_{\mathsf{IB\text{-}HPS},\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} |\Pr[\mathcal{A} \text{ wins}] - \frac{1}{2}|$. We require that $\mathsf{Adv}^{\mathsf{VI\text{-}IND}}_{\mathsf{IB\text{-}HPS},\mathcal{A}}(\lambda) \le \mathsf{negl}(\lambda)$.

III. Smoothness. Other than properties I and II, we will need one additional information theoretic property. Essentially, we want to ensure that there are many possibilities for the decapsulation of an *invalid* ciphertext, which are left undetermined by the public parameters of the system.

Definition 2.2 (Smooth IB-HPS). *We say an* IB-HPS *is* **smooth** *if, for any* fixed values *of* $\mathsf{mpk},\mathsf{msk}$ *produced by* $\mathtt{Setup}(1^\lambda)$, *any* $\mathsf{id} \in \mathcal{ID}$, *we have*

$$\mathbf{SD}((\mathsf{C},k),(\mathsf{C},k')) \le \mathsf{negl}(\lambda)$$

where $\mathsf{C} \leftarrow \mathtt{Encap}^*(\mathsf{id})$, $k \leftarrow \mathtt{Decap}(\mathsf{C},\mathtt{KeyGen}(\mathsf{id},\mathsf{msk}))$ *and* $k' \xleftarrow{\$} \mathcal{K}$.

2.3 Extractors

In our constructions we will use some of the following notions and primitives. For a detailed treatment see [31, 13].

The min-entropy of a random variable X is defined as $\mathbf{H}_\infty(X) = -\log(\max_x \Pr[X = x])$. We will mainly use the average min-entropy of a random variable X conditioned on another random variable Y. This is defined as

$$\tilde{\mathbf{H}}_\infty(X|Y) = -\log\left(\mathbb{E}_{y\leftarrow Y}\left[\max_x \Pr\left[X = x | Y = y\right]\right]\right)$$

where $\mathbb{E}_{y\leftarrow Y}$ denotes the expected value over $y \leftarrow Y$.

Definition 2.3. *We say that an efficient randomized function* $\mathsf{Ext} : \mathcal{K} \to \{0,1\}^\nu$ *is an (average-case)* (μ,ϵ)-extractor *(for space* $\mathcal{K}$*) if for all* X, Z *such that* X *is distributed over* $\mathcal{K}$ *and* $\widetilde{\mathbf{H}}_\infty(X|Z) \ge \mu$, *we get*

$$\mathbf{SD}((Z, S, \mathsf{Ext}(X;S)),(Z,S,U_\nu)) \le \epsilon,$$

where S *denotes the coins of* Ext *(called the* seed*), and* U_ν *is the uniform distribution over* $\{0,1\}^\nu$.

An extractor can be used to extract uniform randomness out of a weakly-random value which is only assumed to have sufficient min-entropy μ. The famous leftover hash lemma (see [13]) states that one can have efficient extractors capable of extracting almost μ (nearly) uniform random bits. We also notice that, with a proper implementation, the resulting extractor is at least as efficient as a cryptographic hash function, making its cost negligible for our purposes.

LEMMA 2.4. *For any output length* $\nu \leq \mu - 2\log(1/\epsilon) - 1$, *there exists very efficient* (μ, ϵ)*-extractor* $\mathsf{Ext} : \mathcal{K} \to \{0,1\}^\nu$.

2.4 Leakage-Resilient IBE from IB-HPS

Alwen *et al.* [2] showed how to convert a smooth IB-HPS, given by algorithms $(\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Encap}, \mathtt{Encap}^*, \mathtt{Decap})$, into an ℓ-leakage resilient IBE. We assume that the encapsulated key space $\mathcal{K}$ has size 2^μ, and will use $(\mu-\ell, \epsilon)$-extractor $\mathsf{Ext} : \mathcal{K} \to \{0,1\}^\nu$, where $\epsilon \leq \mathsf{negl}(\lambda)$.

The resulting IBE will have the same identity set $\mathcal{ID}$ and the same $\mathtt{Setup}, \mathtt{KeyGen}$ algorithms,[2] but will operate on the message space $\mathcal{M} = \{0,1\}^\nu$. The $\mathtt{Encrypt}, \mathtt{Decrypt}$ algorithms are defined as follows:

$\mathtt{Encrypt}(\mathsf{id}, M)$: Choose $(c_1, k) \leftarrow \mathtt{Encap}(\mathsf{id})$ and seed s for Ext, and let $c_2 = \mathsf{Ext}(k; s) \oplus M$. Output $\mathsf{C} = (c_1, s, c_2)$.

$\mathtt{Decrypt}(\mathsf{C}, \mathsf{sk_{id}})$: Parse $\mathsf{C} = (c_1, s, c_2)$ and then compute $k = \mathtt{Decap}(c_1, \mathsf{sk_{id}})$. Output $M = c_2 \oplus \mathsf{Ext}(k; s)$.

Alwen *et al.* [2] showed the following intuitive result:

THEOREM 2.5 ([2]). *Assume that an* IB-HPS *is smooth and that the size of the key set* $\mathcal{K}$ *is* $|\mathcal{K}| = 2^\mu$. *Let* $\mathsf{Ext} : \mathcal{K} \to \{0,1\}^\nu$ *be a* $(\mu - \ell, \epsilon)$*-extractor for some* $\epsilon \leq \mathsf{negl}(\lambda)$. *Then the above transformation produces an* ℓ*-leakage-resilient IBE. In particular, by Lemma 2.4, the resulting IBE can achieve relative leakage arbitrarily close to* $\hat{\alpha} \approx \mu/\hat{\mu}$, *where* $\hat{\mu}$ *is the bit size of individual secret key* $\mathsf{sk_{id}}$.

2.5 Bilinear Groups and Assumptions

We assume the existence of a group generator algorithm $\mathcal{G}(1^\lambda)$ which, on input 1^λ, outputs a tuple $(p, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot))$ where groups $\mathbb{G}, \mathbb{G}_T$ are of prime order $p = \Theta(2^\lambda)$ which admit an efficiently computable non-degenerate bilinear map[3] $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Similarly, we also assume the existence of an algorithm $\mathcal{G}'(1^\lambda)$ which outputs a tuple $(N, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot))$ where groups $\mathbb{G}, \mathbb{G}_T$ are of composite order $N = p_1p_2p_3$, p_1, p_2, p_3 are three prime numbers of magnitude $\Theta(2^\lambda)$, and $e(\cdot,\cdot)$ is defined similarly as in $\mathcal{G}(1^\lambda)$. We further require that the group operations in $\mathbb{G}$ and $\mathbb{G}_T$ as well as $e(\cdot,\cdot)$ are computable in polynomial time with respect to λ, and the group descriptions of $\mathbb{G}$ and $\mathbb{G}_T$ include generators of the respective cyclic groups. Denote by $\mathbb{G}_j$ the subgroup of $\mathbb{G}$ of order j. Generators of all subgroups of $\mathbb{G}$ can be generated in polynomial time with the factorization of the group order.

We review four problems. The first one is about prime order group and we use it to prove security of the first two systems in Sections 3, 4. The others are related to composite-order groups which are for the last system in Section 5.

[2]Assuming Ext is publicly known; otherwise, mpk will also include the description of Ext.

[3]We require that for every generator $g \in \mathbb{G}$ we have that $e(g,g) \neq 1$ and for every $a, b \in \mathbb{Z}_p : e(g^a, g^b) = e(g,g)^{ab}$.

ASSUMPTION 2.6 (DBDH). *Decisional Bilinear Diffie-Hellman problem is, given* $\mathsf{PP} = (p, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot)) \leftarrow \mathcal{G}(1^\lambda)$, $D = (g, g^x, g^y, g^z) \in \mathbb{G}^4$ *and* $T_\nu \in \mathbb{G}_T$, *distinguish between* $\nu = 0$ *or* $\nu = 1$, *where* $T_0 = e(g,g)^{xyz}$ *and* $T_1 \xleftarrow{\$} \mathbb{G}_T$. *The advantage of an algorithm* $\mathcal{A}$ *in solving DBDH is defined as*

$$\mathsf{Adv}^{\mathsf{DBDH}}_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(\mathsf{PP}, D, T_0) = 0] - \Pr[\mathcal{A}(\mathsf{PP}, D, T_1) = 0]|.$$

We say that the DBDH assumption holds if for all PPT adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{DBDH}}_{\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda)$.

ASSUMPTION 2.7 (ASSUMPTIONS 1, 2, 3). *We first define the variables* $\mathsf{PP}' = (N, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot)) \leftarrow \mathcal{G}'(1^\lambda)$, $\beta, z \xleftarrow{\$} \mathbb{Z}_N$, $g, X_1 \xleftarrow{\$} \mathbb{G}_{p_1}$, $X_2, Y_2, Z_2 \xleftarrow{\$} \mathbb{G}_{p_2}$, $X_3, Y_3 \xleftarrow{\$} \mathbb{G}_{p_3}$,

$$\begin{aligned}
&D^{(1)} = (g, X_3),\\
&T_0^{(1)} \xleftarrow{\$} \mathbb{G}_{p_1p_2}, \qquad T_1^{(1)} \xleftarrow{\$} \mathbb{G}_{p_1},\\
&D^{(2)} = (g, X_1X_2, X_3, Y_2Y_3),\\
&T_0^{(2)} \xleftarrow{\$} \mathbb{G}, \qquad T_1^{(2)} \xleftarrow{\$} \mathbb{G}_{p_1p_3},\\
&D^{(3)} = (g, g^\beta X_2, g^z Y_2, Z_2, X_3),\\
&T_0^{(3)} \leftarrow e(g,g)^{\beta z}, \quad T_1^{(3)} \xleftarrow{\$} \mathbb{G}_T.
\end{aligned}$$

We define $\mathsf{Adv}^{(i)}_{\mathcal{A}}(\lambda)$, *the advantage of an algorithm* $\mathcal{A}$, *to be* $|\Pr[\mathcal{A}(\mathsf{PP}', D^{(i)}, T_0^{(i)}) = 0] - \Pr[\mathcal{A}(\mathsf{PP}', D^{(i)}, T_1^{(i)}) = 0]|$. *Assumption* i *holds if for all PPT* $\mathcal{A}$, $\mathsf{Adv}^{(i)}_{\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda)$.

3. OUR FIRST SYSTEM

Our first system is similar to Boneh-Boyen IBE [4] and its security is based on the same assumption (DBDH).

We "tag" a user secret key with an integer tag t by introducing a factor of $g^{\beta t}$ ($\beta \in \mathbb{Z}_p$ is a new secret parameter) to the term which already has g^α ($\alpha \in \mathbb{Z}_p$ is an existing secret parameter). Intuitively, for an attacker who only gets one secret key for each identity, deriving a new key of a different tag requires the knowledge of g^α or g^β. To offset the effect of $g^{\beta t}$ in decryption, the ciphertext requires a new component of $e(g,g)^{\beta z}$ where $z \in \mathbb{Z}_p$ is its randomness. The same design principle is used for all systems in this paper.

3.1 Construction

We now present the system which is denoted by $\mathsf{CDRW^{BB}}$.

$\mathsf{CDRW^{BB}}$

$\mathtt{Setup}(1^\lambda)$: Let $(p, \mathbb{G} = \langle g \rangle, \mathbb{G}_T, e(\cdot,\cdot)) \leftarrow \mathcal{G}(1^\lambda)$. Set $\mathsf{mpk} = (p, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot), g, u, h, e(g,g)^\alpha, e(g,g)^\beta)$ where $u, h \xleftarrow{\$} \mathbb{G}$ and $\alpha, \beta \xleftarrow{\$} \mathbb{Z}_p$. Also set $\mathsf{msk} = (g^\alpha, g^\beta)$.

$\mathtt{KeyGen}(\mathsf{id}, \mathsf{msk})$: For $\mathsf{id} \in \mathbb{Z}_p$, choose $t, r \xleftarrow{\$} \mathbb{Z}_p$. Output $\mathsf{sk_{id}} = (s_1, s_2, s_3) = (g^\alpha g^{-\beta t}(u^{\mathsf{id}}h)^r, g^{-r}, t)$.

$\mathtt{Encap}(\mathsf{id})$: Choose $z \xleftarrow{\$} \mathbb{Z}_p$. Output $\mathsf{C} = (c_1, c_2, c_3) = (g^z, (u^{\mathsf{id}}h)^z, e(g,g)^{\beta z})$ and $k = e(g,g)^{\alpha z}$.

$\mathtt{Encap}^*(\mathsf{id})$: Choose $z, z' \xleftarrow{\$} \mathbb{Z}_p$ subject to the constraint $z \neq z'$. Output $\mathsf{C} = (c_1, c_2, c_3) = (g^z, (u^{\mathsf{id}}h)^z, e(g,g)^{\beta z'})$.

$\mathtt{Decap}(\mathsf{C}, \mathsf{sk_{id}})$: Output $e(c_1, s_1)e(c_2, s_2)c_3^{s_3}$.

THEOREM 3.1. *If the DBDH assumption holds,* $\mathsf{CDRW^{BB}}$ *is a smooth* IB-HPS.

PROOF. For *correctness* of decapsulation, we have

$$\begin{aligned}
& e(c_1, s_1)e(c_2, s_2)c_3^{s_3} \\
= \;& e(g^z, g^\alpha g^{-\beta t}(u^{\mathsf{id}}h)^r)e((u^{\mathsf{id}}h)^z, g^{-r})(e(g,g)^{\beta z})^t \\
= \;& e(g, g^{\alpha z} g^{-\beta z t}(u^{\mathsf{id}}h)^{zr})e((u^{\mathsf{id}}h)^{-zr}, g)e(g,g)^{\beta z t} \\
= \;& e(g, g^{\alpha z})e(g, g^{-\beta z t}(u^{\mathsf{id}}h)^{zr})e(g, (u^{\mathsf{id}}h)^{-zr})e(g, g^{\beta z t}) \\
= \;& e(g,g)^{\alpha z}
\end{aligned}$$

For *smoothness*, for any fixed $(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$, consider an invalid ciphertext $(c_1, c_2, c_3) = (g^z, (u^{\mathsf{id}}h)^z, e(g,g)^{\beta z'})$ and any secret key $(s_1, s_2, s_3) = (g^\alpha g^{-\beta t}(u^{\mathsf{id}}h)^r, g^{-r}, t)$, we have

$$\begin{aligned}
& e(c_1, s_1)e(c_2, s_2)c_3^{s_3} \\
= \;& e(g^z, g^\alpha g^{-\beta t}(u^{\mathsf{id}}h)^r)e((u^{\mathsf{id}}h)^z, g^{-r})(e(g,g)^{\beta z'})^t \\
= \;& e(g, g^{\alpha z} g^{-\beta z t}(u^{\mathsf{id}}h)^{zr})e((u^{\mathsf{id}}h)^{-zr}, g)e(g,g)^{\beta z' t} \\
= \;& e(g, g^{\alpha z})e(g, g^{-\beta z t})e(g, g^{\beta z' t}) \\
= \;& e(g,g)^{\alpha z} e(g,g)^{t\beta(z'-z)}
\end{aligned}$$

So, for any fixed C output by $\mathtt{Encap}^*(\mathsf{id})$, the distribution of $\mathtt{Decap}(\mathsf{C}, \mathsf{sk}_{\mathsf{id}})$, over a uniform $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathtt{KeyGen}(\mathsf{msk}, \mathsf{id})$, is uniform over $\mathbb{G}_T$. This implies smoothness.

The most delicate property of valid/invalid ciphertext indistinguishability will be done in Section 3.2 below. Here we just comment on the relative leakage $\hat{\alpha} = \mu/\hat{\mu}$ as explained in Theorem 2.5. The encapsulated-key size is $\mu = \log p$ and the bit size of individual secret key is $\hat{\mu} = 3\log p + O(1)$. So the leakage allowed is slightly less than 1/3. □

3.2 Ciphertext Indistinguishability

Using the original Boneh-Boyen system naturally allows the cancellation of the master key component g^α in the simulation of secret keys for identities different than id^*. However in our setting the simulator has to create one secret key for id^* which is not possible in the original simulation. In order to do this we added one more parameter β and the tag t. Now the DBDH parameters x and y are embedded in $\beta = xy$. The tag is used as a "trapdoor" for the simulator of our proofs to relate α to β in a way hidden from any attacker without knowing β. Specifically, he picks a known tag t^* and sets $\alpha = \beta t^* + \tilde{\alpha}$. Knowledge of t^* allows him to create secret keys of the challenge identity id^* tagged only with t^* since the resulting g^{xyt^*} from g^α is cancelled by the g^{-xyt^*} term from $g^{-\beta t^*}$. The original trick of Boneh-Boyen still works for identities other than id^*. This key will seem random to the attacker; as if it was sampled from the entire space of the secret keys for id^* since the dependence of α on x, y and t^* is oblivious to the attacker.

The new term $e(g,g)^{xyz}$ in the ciphertext will serve as the challenge term of DBDH. Valid/invalid ciphertext indistinguishability comes from this term which is transformed to a random term in case a non-valid DBDH term is given. The same intuition holds for all three systems in this paper.

PROOF (CONTINUED FROM SECTION 3.1). $\mathcal{B}$ is given the tuple $(g, g^x, g^y, g^z, T_\nu)$ from a DBDH challenge. According to the game for selective security, $\mathcal{A}$ gives to $\mathcal{B}$ an identity id^* that he wishes to attack.

For setup, $\mathcal{B}$ sets $u = g^x$, $h = (g^x)^{-\mathsf{id}^*} g^{\tilde{h}}$, implicitly sets $\beta = xy$ and computes $e(g,g)^\beta = e(g^x, g^y)$. Finally, he implicitly sets $\alpha = \beta t^* + \tilde{\alpha}$ and compute $e(g,g)^\alpha = e(g^x, g^y)^{t^*} e(g,g)^{\tilde{\alpha}}$, where all new variables such as $\tilde{h}, t^*, \tilde{\alpha}$ are chosen uniformly at random from $\mathbb{Z}_p$ from this point on.

We argue that the variables α, h are properly distributed because $\mathcal{B}$ picks the random exponents $\tilde{\alpha}, \tilde{h}$ uniformly at random. The elements g, x, y are supposed to be picked uniformly at randomly by the DBDH challenger from their respective groups, which makes the variables u, β properly distributed in particular. Hence, the view of $\mathcal{A}$ is completely legitimate when $\mathcal{B}$ responds with the public parameter

$$\mathsf{mpk} = (p, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot), g, u, h, e(g,g)^\alpha, e(g,g)^\beta).$$

For test stage 1, $\mathcal{B}$ has to compute secret keys for all identities. For queries on $\mathsf{id} \neq \mathsf{id}^*$, $\mathcal{B}$ sets $s_3' = t^* - \tilde{t}$ and implicitly sets $r = -y\tilde{t}/(\mathsf{id} - \mathsf{id}^*) + \tilde{r}$, which is properly distributed since $\tilde{r}$ is chosen uniformly at random. Then he computes

$$\begin{aligned}
s_1' &= g^\alpha g^{-\beta t}(u^{\mathsf{id}}h)^r \\
&= g^{xyt^* + \tilde{\alpha}} \cdot (g^{-xyt^*} g^{xy\tilde{t}}) \cdot (g^{x(\mathsf{id}-\mathsf{id}^*)} g^{\tilde{h}})^{\frac{-y\tilde{t}}{(\mathsf{id}-\mathsf{id}^*)}} \cdot (u^{\mathsf{id}}h)^{\tilde{r}} \\
&= g^{\tilde{\alpha}} (g^y)^{-\tilde{h}\tilde{t}/(\mathsf{id}-\mathsf{id}^*)} (u^{\mathsf{id}}h)^{\tilde{r}} \\
s_2' &= g^{-r} = (g^y)^{\tilde{t}/(\mathsf{id}-\mathsf{id}^*)} g^{-\tilde{r}}
\end{aligned}$$

and responds with $\mathsf{sk}_{\mathsf{id}} = (s_1', s_2', s_3')$.

For the query on id^*, he sets $s_3^* = t^*$ and picks an exponent r. Since t^* was chosen randomly in the setup, s_3^* is properly distributed. Then he computes

$$s_1^* = g^\alpha g^{-\beta t^*}(u^{\mathsf{id}^*}h)^r = g^{\beta t^* + \tilde{\alpha}} \cdot g^{-\beta t^*} \cdot (u^{\mathsf{id}^*}h)^r = g^{\tilde{\alpha}}(u^{\mathsf{id}^*}h)^r,$$

and $s_2^* = g^{-r}$. The query is answered by $\mathsf{sk}_{\mathsf{id}^*} = (s_1^*, s_2^*, s_3^*)$.

By choosing different $\tilde{t}$'s $\mathcal{B}$ can calculate many keys on id with different tags since $t = t^* - \tilde{t}$, but he can generate key with only one tag for id^* since $u^{\mathsf{id}^*}h = g^{x\mathsf{id}^*} g^{-x\mathsf{id}^*} g^{\tilde{h}} = g^{\tilde{h}}$.

For challenge, $\mathcal{B}$ returns $(c_1, c_2, c_3) = (g^z, (g^z)^{\tilde{h}}, T_\nu)$.

For test stage 2, $\mathcal{B}$ calculates the secret keys as he did in test stage 1. Eventually, $\mathcal{A}$ outputs a guess φ', then $\mathcal{B}$ returns φ'.

It is easy to see that c_2 is valid by recalling $u^{\mathsf{id}^*}h = g^{\tilde{h}}$. With $\beta = xy$, it is easy to see that c_3 is valid if $T_\nu = e(g,g)^{xyz}$. This (perfectly) corresponds to the distribution of a valid ciphertext. On the other hand if $T_\nu \overset{\$}{=\!\leftarrow} \mathbb{G}_T$, then $c_3 = e(g,g)^{\beta z'}$ for a random z', independent of z which is $1/p$ statistically close to the distribution of invalid ciphertexts output by $\mathtt{Encap}^*(\mathsf{id}^*)$. We thus proved that the advantage of $\mathcal{B}$ in breaking the DBDH assumption is negligibly close to the advantage of $\mathcal{A}$ in the selective-ID valid/invalid indistinguishability game. □

4. OUR SECOND SYSTEM

Now we modify $\mathsf{CDRW}^{\mathsf{BB}}$ in a way similar to [35] to get an adaptively-secure identity-based hash proof system. We assume that all identities are B-bit vectors, and we denote the i-th bit of id by id_i. The only change is that we use Waters hash $u_0 \prod_{i=1}^{B} u_i^{\mathsf{id}_i}$ for identity $\mathsf{id} = (\mathsf{id}_1, \mathsf{id}_2, \cdots, \mathsf{id}_B) \in \{0,1\}^B$, where B is a polynomial in the security parameter λ, instead of Boneh-Boyen hash $u^{\mathsf{id}}h$ for $\mathsf{id} \in \mathbb{Z}_p$.

$\mathsf{CDRW^{W}}$

$\mathtt{Setup}(1^\lambda)$: Let $(p, \mathbb{G} = \langle g \rangle, \mathbb{G}_T, e(\cdot,\cdot)) \leftarrow \mathcal{G}(1^\lambda)$. Pick $\alpha, \beta \xleftarrow{\$} \mathbb{Z}_p$ and $u_0, \cdots, u_B \xleftarrow{\$} \mathbb{G}$. Set $\mathsf{msk} = (g^\alpha, g^\beta)$. Set $\mathsf{mpk} = (p, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot), g, u_0, \cdots, u_B, e(g,g)^\alpha, e(g,g)^\beta)$.

$\mathtt{KeyGen}(\mathsf{id}, \mathsf{msk})$: For $\mathsf{id} \in \{0,1\}^B$, choose $t, r \xleftarrow{\$} \mathbb{Z}_p$. Output $\mathsf{sk_{id}} = (s_1, s_2, s_3) = (g^\alpha g^{-\beta t}(u_0 \prod_{i=1}^B u_i^{\mathsf{id}_i})^r, g^{-r}, t)$.

$\mathtt{Encap}(\mathsf{id})$: Choose $z \xleftarrow{\$} \mathbb{Z}_p$. Output $\mathsf{C} = (c_1, c_2, c_3) = (g^z, (u_0 \prod_{i=1}^B u_i^{\mathsf{id}_i})^z, e(g,g)^{\beta z})$ and $k = e(g,g)^{\alpha z}$.

$\mathtt{Encap}^*(\mathsf{id})$: Choose $z, z' \xleftarrow{\$} \mathbb{Z}_p$ where $z \neq z'$. Output $\mathsf{C} = (c_1, c_2, c_3) = (g^z, (u_0 \prod_{i=1}^B u_i^{\mathsf{id}_i})^z, e(g,g)^{\beta z'})$.

$\mathtt{Decap}(\mathsf{C}, \mathsf{sk_{id}})$: Output $e(c_1, s_1)e(c_2, s_2)c_3^{s_3}$.

THEOREM 4.1. *If the DBDH assumption holds,* $\mathsf{CDRW^W}$ *is a smooth* IB-HPS*.*

PROOF. Correctness and smoothness are easy to see. The relative amount of leakage is slightly less than 1/3 of the secret key for the same reasons as in $\mathsf{CDRW^{BB}}$. The proof for valid/invalid ciphertext indistinguishability follows the original security proof [35] modified to our IB-HPS system and security game. Details can be found in the full version [9]. □

5. OUR THIRD SYSTEM

Our third system is based on the dual encryption system of Lewko-Waters [28], designed in order to achieve full security with smaller public parameters size. The transformation we apply is similar to the one we used in Boneh-Boyen system.

5.1 Construction

Now we present $\mathsf{CDRW^{LW}}$, the last IB-HPS in this paper.

$\mathsf{CDRW^{LW}}$

$\mathtt{Setup}(1^\lambda)$: Let $(N = p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot)) \leftarrow \mathcal{G}'(1^\lambda)$. Set $\mathsf{mpk} = (N, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot), g, u, h, e(g,g)^\alpha, e(g,g)^\beta)$ where g is a generator of $\mathbb{G}_{p_1}$, $u, h \xleftarrow{\$} \mathbb{G}_{p_1}$ and $\alpha, \beta \xleftarrow{\$} \mathbb{Z}_N$. Set $\mathsf{msk} = (g^\alpha, g^\beta, X_3)$ where X_3 is a generator of $\mathbb{G}_{p_3}$.

$\mathtt{KeyGen}(\mathsf{id}, \mathsf{msk})$: For $\mathsf{id} \in \mathbb{Z}_N$, choose $t, r, \rho, \rho' \xleftarrow{\$} \mathbb{Z}_N$. Output $\mathsf{sk_{id}} = (s_1, s_2, s_3) = (g^\alpha g^{-\beta t}(u^{\mathsf{id}}h)^r X_3^\rho, g^{-r}X_3^{\rho'}, t)$.

$\mathtt{Encap}(\mathsf{id})$: Choose $z \xleftarrow{\$} \mathbb{Z}_N$. Output $\mathsf{C} = (c_1, c_2, c_3) = (g^z, (u^{\mathsf{id}}h)^z, e(g,g)^{\beta z})$ and $k = e(g,g)^{\alpha z}$.

$\mathtt{Encap}^*(\mathsf{id})$: Choose $z, z' \xleftarrow{\$} \mathbb{Z}_N$ subject to the constraint $z \neq z'$. Output $\mathsf{C} = (c_1, c_2, c_3) = (g^z, (u^{\mathsf{id}}h)^z, e(g,g)^{\beta z'})$.

$\mathtt{Decap}(\mathsf{C}, \mathsf{sk_{id}})$: Output $e(c_1, s_1)e(c_2, s_2)c_3^{s_3}$.

THEOREM 5.1. *Under Assumptions 1, 2 and 3,* $\mathsf{CDRW^{LW}}$ *is a smooth* IB-HPS*.*

PROOF. Decapsulation of a valid ciphertext gives

$$\begin{aligned} & e(c_1, s_1)e(c_2, s_2)c_3^{s_3} \\ = \; & e(g^z, g^\alpha g^{-\beta t}(u^{\mathsf{id}}h)^r X_3^\rho)e((u^{\mathsf{id}}h)^z, g^{-r}X_3^{\rho'})(e(g,g)^{\beta z})^t \\ = \; & e(g,g)^{\alpha z}e(g,g)^{-\beta z t}e(g, u^{\mathsf{id}}h)^{zr}e(g, X_3)^{z\rho} \\ & e(u^{\mathsf{id}}h, g)^{-zr}e(u^{\mathsf{id}}h, X_3)^{z\rho'}e(g,g)^{\beta z t} \\ = \; & e(g,g)^{\alpha z}e(g,g)^{-\beta z t}e(g, u^{\mathsf{id}}h)^{zr}e(g, u^{\mathsf{id}}h)^{-zr}e(g,g)^{\beta z t} \\ = \; & e(g,g)^{\alpha z} \end{aligned}$$

The terms $e(g, X_3)^{z\rho}$ $e(u^{\mathsf{id}}h, X_3)^{z\rho'}$ disappear since g, u, h are in $\mathbb{G}_{p_1}$ and X_3 is in $\mathbb{G}_{p_3}$. This proves *correctness*. On the other hand, decapsulation of an invalid ciphertext gives

$$e(c_1, s_1)e(c_2, s_2)c_3^{s_3} = e(g,g)^{\alpha z}e(g,g)^{t\beta(z'-z)}.$$

So, for any fixed C output by $\mathtt{Encap}^*(\mathsf{id})$, the distribution of $\mathtt{Decap}(\mathsf{C}, \mathsf{sk_{id}})$ is uniform over $\mathbb{G}_T$. This implies *smoothness*.

The security proof for valid/invalid ciphertext indistinguishability is much more complicated than that of $\mathsf{CDRW^{BB}}$ since we have to move from the original security game to a game where all secret keys and the ciphertext have a specific *semi-functional* form. We dedicate Section 5.2 for that.

We observe that the encapsulated-key size is $\log |\mathbb{G}_{T_{p_1}}|$ where $\mathbb{G}_{T_{p_1}}$ denotes the subgroup of $\mathbb{G}_T$ of order p_1. If we assume that elements in the composite order group $\mathbb{G}_T$ can be represented with 3λ bits, we get that the fraction of the secret key that can be leaked is slightly less than 1/9. □

5.2 Ciphertext Indistinguishability

To prove valid/invalid ciphertext indistinguishability we will use the additional structures, defined in [28], of semi-functional ciphertexts and semi-functional keys. The keys generated by $\mathtt{KeyGen}$ and the ciphertexts generated by $\mathtt{Encap}$ and $\mathtt{Encap}^*$ will be referred to as normal.

DEFINITION 5.2. *To create a semi-functional key, firstly a normal key* (s_1, s_2, s_3) *is created, then a random element* $X_2 \xleftarrow{\$} \mathbb{G}_{p_2}$ *and a random exponent* $\theta_k \xleftarrow{\$} \mathbb{Z}_N$ *are chosen. The semi-functional key is* $(s_1', s_2', s_3') = (s_1 X_2^{\theta_k}, s_2 X_2^{-1}, s_3)$.

DEFINITION 5.3. *To create a semi-functional ciphertext, firstly a normal ciphertext* (c_1, c_2, c_3) *is created, then a random element* $Y_2 \xleftarrow{\$} \mathbb{G}_{p_2}$ *and a random exponent* $\theta_c \xleftarrow{\$} \mathbb{Z}_N$ *are chosen. The semi-functional ciphertext is* $(c_1', c_2', c_3') = (c_1 Y_2, c_2 Y_2^{\theta_c}, c_3)$.

Using a semi-functional key to decrypt a semi-functional ciphertext will result in something depends on θ_k and θ_c.

$$\begin{aligned} & e(c_1', s_1')e(c_2', s_2')(c_3')^{s_3'} \\ & e(c_1, s_1)e(c_2, s_2)(c_3)^{s_3}e(Y_2, X_2^{\theta_k})e(Y_2^{\theta_c}, X_2^{-1}) \\ = \; & e(g,g)^{\alpha z}e(Y_2, X_2)^{(\theta_k - \theta_c)} \end{aligned}$$

The proof of security relies on Assumptions 1,2,3 of Section 2.5. We will define a sequence of games and use them in a hybrid proof of security.

VCI. The first game is the normal security game as defined in Section 2.2 which the challenger chooses $b = 0$, i.e., a (normal) valid ciphertext is given to the adversary.

Res. The next game is the same except that all identities the attacker queries are not equal to the challenge identity id^* mod p_2. That means that when the attacker gives an id^* such that for some id in test stage 1 or 2, $\mathsf{id}^* = \mathsf{id}$ mod p_2 the challenger picks a random bit $\varphi \xleftarrow{\$} \{0,1\}$ and the attacker succeeds if $\varphi = 0$. We will retain this restriction in all subsequent games.

KG_i. We denote by $L(\lambda)$ the maximum number of different identities used in secret-key queries. Then for each $i \in [L-1] \cup \{0\}$ we define the game KG_i to be like the Res security game but the ciphertext is semi-functional and the generated keys of the first i identities are semi-functional excluding the key of i^*-th queried identity where i^* is an index randomly selected from $[L]$. If the i^*-th queried identity is not the challenge identity, the challenger picks a random bit $\varphi \xleftarrow{\$} \{0,1\}$ and the attacker succeeds if $\varphi = 0$. We will retain this restriction in all subsequent games. In game KG_0 all keys are normal and the ciphertext is semi-functional. In game KG_{L-1} all generated keys except the key of the challenge identity are semi-functional.

SICI. The difference of this game from the game KG_K is that the challenge ciphertext is a semi-functional invalid ciphertext, i.e., it is first generated by `Encap`*, then is converted to a semi-functional one as previously described.

KG'_i. For each $i \in [L-1] \cup \{0\}$, we define the game KG'_i to be like the KG_i security game but the ciphertext is invalid. In game KG'_{L-1} all keys except the i^*-th queried identity are semi-functional and the ciphertext is semi-functional and invalid, i.e., $\mathsf{KG}'_{L-1} = \mathsf{SICI}$. In game KG'_0 all keys are normal, and the ciphertext is semi-functional and invalid.

ICI. The final game is the normal security game as defined in Section 2.2 which the challenger chooses $b = 1$, i.e., a (normal) invalid ciphertext is given to the adversary.

The advantages of all algorithms in the above games are defined in the same way as $\mathsf{Adv}^{\mathsf{VI\text{-}IND}}$. In the following lemmas we will prove that all of these games are indistinguishable by PPT attackers under the assumptions in Section 2.5.

LEMMA 5.4. *Suppose there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}^{\mathsf{VCI}}_{\mathcal{A}} - \mathsf{Adv}^{\mathsf{Res}}_{\mathcal{A}} = \epsilon$, we can build a PPT algorithm $\mathcal{B}$ with advantage at least ϵ in breaking either Assumption 1 or Assumption 2.*

PROOF. In both assumptions simulator $\mathcal{B}$ is given g and X_3, $\mathcal{B}$ can then "simulate" $\mathsf{CDRW}^{\mathsf{LW}}$ as in the real world by picking uniformly at random exponents $x, y, \alpha, \beta \xleftarrow{\$} \mathbb{Z}_N$ and setting $u = g^x, h = g^y$. Since he knows the master secret key (α, β, X_3), he can generate secret keys for all identities.

According to the lemma's assumption, $\mathcal{A}$ will query for identities $\mathsf{id} \neq \mathsf{id}^* \bmod N$ and $\mathsf{id} = \mathsf{id}^* \bmod p_2$ with probability ϵ. This means that in the end $\mathcal{B}$ can compute a nontrivial factor of $N = p_1p_2p_3$ by calculating $Q = N/\gcd(\mathsf{id} - \mathsf{id}^*, N)$ where $\gcd(\cdot,\cdot)$ denotes the function for the greatest common divisor. To see, it is true that with probability ϵ:

$$\textit{Case 1: } Q = p_1 \text{ or } Q = p_1p_3 \qquad \textit{Case 2: } Q = p_3$$

One of the two cases occurs with probability at least $\epsilon/2$. In case 1, $\mathcal{B}$ breaks Assumption 1 by raising the challenge term T_ν to Q. If $\nu = 1$ then $T_\nu^Q = 1$. Otherwise $T_\nu^Q \neq 1$.

In case 2, $\mathcal{B}$ tests if $e\left((Y_2Y_3)^Q, T_\nu\right) = 1$. Note that $Y_3^{p_3} = 0$. If T_ν contains no p_2 part, i.e., when $\nu = 1$, this pairing equals 1. Otherwise $T_\nu \in \mathbb{G}$ and $\nu = 0$. □

LEMMA 5.5. *Suppose there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}^{\mathsf{Res}}_{\mathcal{A}} - \mathsf{Adv}^{\mathsf{KG}_0}_{\mathcal{A}} = \epsilon$, we can build a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 1.*

PROOF. As in Lemma 5.4, $\mathcal{B}$ is given g and X_3 and he can simulate a "full" version of $\mathsf{CDRW}^{\mathsf{LW}}$. In particular, he sets $u = g^x, h = g^y$ which makes a normal ciphertext (c_1, c_2, c_3) for id satisfies the relation $c_2 = c_1^{x\mathsf{id}+y}$. In the challenge stage, $\mathcal{B}$ responds with the following ciphertext using the challenge term T_ν after received the challenge identity id^*:

$$(c_1^*, c_2^*, c_3^*) = (T_\nu, T_\nu^{x\mathsf{id}^*+y}, e(T_\nu, g)^\beta).$$

If $T_\nu \in \mathbb{G}_{p_1}$, $\mathcal{A}$ plays game Res since $T_\nu = g^z$ for some $z \in \mathbb{Z}_N$, and hence we get a normal ciphertext for id^*. If $T_\nu \in \mathbb{G}_{p_1p_2}$, T_ν can be written as g^zX_2 for some $X_2 \in \mathbb{G}_{p_2}$ with $e(X_2, g) = 0$, then the above is a semi-functional ciphertext with $\theta_c = x\mathsf{id}^* + y$ and $\mathcal{A}$ plays game KG_0. The value of θ_c modulo p_2 is not correlated with the values of x and y modulo p_1 so this is correctly distributed. Hence, if $\mathcal{B}$ answers $\nu' = 0$ when $\mathcal{A}$ succeeds, he has advantage ϵ in breaking Assumption 1. □

LEMMA 5.6. *Suppose there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}^{\mathsf{KG}_{i-1}}_{\mathcal{A},\mathsf{CDRW}^{\mathsf{LW}}} - \mathsf{Adv}^{\mathsf{KG}_i}_{\mathcal{A},\mathsf{CDRW}^{\mathsf{LW}}} = \epsilon$, we can build a PPT algorithm $\mathcal{B}$ with advantage at least ϵ/L in breaking Assumption 2.*

PROOF. Algorithm $\mathcal{B}$ has to create semi-functional keys for some identities excluding the challenge identity. Since he does not know id^* before the challenge stage, he picks uniformly at random an index $i^* \xleftarrow{\$} [L]$ as a guess for the challenge identity. With probability $1/L$, the guess is correct. Notice that according to the assumptions on how the adversaries work the secret key of the challenge identity has always been asked. For the i^*-th queried identity the secret key is normal, regardless of its position.

As before, $\mathcal{B}$ sets $u = g^x$ and $h = g^y$, and normal keys can always be generated using the $\mathsf{msk} = (\alpha, \beta, X_3)$. For the first $i-1$ secret-key queries responds with a properly distributed semi-functional key by:

$$s_1 = g^\alpha g^{-\beta t}(u^{\mathsf{id}}h)^r(Y_2Y_3)^\rho$$
$$s_2 = g^{-r}(Y_2Y_3)^{\rho'}X_3{}^{\rho''} \quad s_3 = t$$

where id is the queried identity and $t, r, \rho, \rho', \rho'' \xleftarrow{\$} \mathbb{Z}_N$.

For the i-th queried identity he uses the challenge term:

$$s_1 = g^\alpha g^{-\beta t}T_\nu^{\theta_k}X_3{}^\rho$$
$$s_2 = T_\nu^{-1} \quad s_3 = t$$

where $t, \rho \xleftarrow{\$} \mathbb{Z}_N$ and $\theta_k = x\mathsf{id}+y$. For the remaining queries $\mathcal{B}$ creates normal secret keys using his master secret key.

At the challenge stage $\mathcal{A}$ responds with the challenge identity id^*. If $\mathcal{B}$ did not make a correct guess for the challenge identity, he aborts at this point. Otherwise he returns the following challenge ciphertext:

$$(c_1^*, c_2^*, c_3^*) = ((X_1X_2), (X_1X_2)^{x\mathsf{id}^*+y}, e((X_1X_2), g)^\beta)$$

which is a properly distributed semi-functional ciphertext with $\theta_c = x\mathsf{id}^* + y$ since $e(X_2, g) = 0$.

This is where we use our modular restriction that all identities are different to the challenge identity modulo p_2. Since for all queries $\mathsf{id} \neq \mathsf{id}^* \bmod p_2$, we have that the $\theta_k = x\mathsf{id}+y$ for the i-th identity and the $\theta_c = x\mathsf{id}^* + y$ seem randomly distributed to $\mathcal{A}$ modulo p_2. This relationship is the reason that we can not create a semi-functional secret key for the challenge identity. Then we would have $\theta_c = \theta_k \bmod p_2$ and obviously they would not be properly distributed, i.e., random. For test stage 2 $\mathcal{B}$ answers with normal secret keys.

Notice that if $T_\nu \in \mathbb{G}_{p_1p_3}$ then the secret key of the i-th queried identity is normal. Thus $\mathcal{A}$ played game KG_{i-1}. If $T_\nu \in \mathbb{G}$ then this secret key is semi-functional and $\mathcal{A}$ played the KG_i game. Therefore if $\mathcal{B}$ answers $\nu' = 0$ when $\mathcal{A}$ succeeds, he breaks Assumption 2 with probability ϵ/L. □

LEMMA 5.7. *Suppose there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{KG}_L} - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{SICI}} = \epsilon$, we can build a PPT algorithm $\mathcal{B}$ with advantage ϵ/L in breaking Assumption 3.*

PROOF. According to Assumption 3, $\mathcal{B}$ first receives

$$(g, g^{\beta}X_2, g^{z}Y_2, Z_2, X_3, T_{\nu}).$$

It chooses random exponents $x, y, t^*, \tilde{\alpha} \xleftarrow{\$} \mathbb{Z}_N$ and sets implicitly $\alpha = t^*\beta + \tilde{\alpha}$. Notice that he does not know the master secret key. He calculates the public parameters as:

$$\begin{aligned} u &= g^{x} \quad e(g,g)^{\beta} = e(g^{\beta}X_2, g) \\ h &= g^{y} \quad e(g,g)^{\alpha} = \left(e(g,g)^{\beta}\right)^{t^*} e(g,g)^{\tilde{\alpha}} \end{aligned}$$

and sends them to $\mathcal{A}$. As in the other proofs $\mathcal{B}$ guesses the challenge identity by picking a random $i^* \in [L]$. With probability $1/L$ the guess is correct.

In test stage 1, when $\mathcal{A}$ asks for a key on identity $\mathsf{id} \neq \mathsf{id}^{(i^*)}$ $\mathcal{B}$ generates the following semi-functional keys. It picks random exponents $\tilde{t}, r, \rho, \rho', \rho'', \rho''' \xleftarrow{\$} \mathbb{Z}_N$ and computes:

$$\begin{aligned} s_1' &= (g^{\beta}X_2)^{\tilde{t}} g^{\tilde{\alpha}} (u^{\mathsf{id}}h)^{r} X_3^{\rho} Z_2^{\rho'''} \\ s_2' &= g^{-r} X_3^{\rho'} Z_2^{\rho''} \quad s_3' = t^* - \tilde{t} \end{aligned}$$

The above is a properly distributed semi-functional secret key, because $g^{\tilde{\alpha}}(g^{\beta})^{\tilde{t}} = g^{\alpha}g^{-\beta s_3'}$.

For the secret key of id^*, $\mathcal{B}$ picks random $r, \rho, \rho' \xleftarrow{\$} \mathbb{Z}_N$ and computes a correctly distributed normal key by:

$$s_1^* = g^{\tilde{\alpha}}(u^{\mathsf{id}^*}h)^{r}X_3^{\rho}, \qquad s_2^* = g^{-r}X_3^{\rho'}, \qquad s_3^* = t^*.$$

$\mathcal{A}$ gives to $\mathcal{B}$ the challenge identity id^*. $\mathcal{B}$ returns

$$(c_1^*, c_2^*, c_3^*) = ((g^{z}Y_2), (g^{z}Y_2)^{x\mathsf{id}^*+y}, T_{\nu}).$$

Since the values of x, y matter only modulo p_1 and θ_c matters only modulo p_2, there is no correlation between them and $\theta_c = x\mathsf{id}^* + y$ seems randomly chosen for any adversary.

If $T_{\nu} = e(g,g)^{\beta z}$, then this is a valid semi-functional ciphertext. Therefore this game is the game KG_L. If $T_{\nu} \in \mathbb{G}_T$, then $\mathcal{B}$ creates an invalid semi-functional ciphertext. This means that $\mathcal{A}$ played the game SICI. Therefore $\mathcal{B}$ breaks Assumption 3 with advantage ϵ/L. □

LEMMA 5.8. *Suppose there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{KG}_i'} - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{KG}_{i-1}'} = \epsilon$, we can build a PPT algorithm $\mathcal{B}$ with advantage ϵ/L in breaking Assumption 2.*

LEMMA 5.9. *Suppose there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{KG}_1'} - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{ICI}} = \epsilon$, we can build a PPT algorithm $\mathcal{B}$ with advantage at least ϵ/L in breaking Assumption 2.*

PROOF. The proofs for Lemmas 5.8 and 5.9 are similar to those for Lemmas 5.6 and 5.5. Note that the c_3 component of the challenge ciphertext can be easily randomized with a random exponent z'. □

THEOREM 5.10. *If Assumptions 1, 2 and 3 hold,* $\mathsf{CDRW}^{\mathsf{LW}}$ *is valid/invalid ciphertext indistinguishable.*

PROOF. Suppose $\epsilon_1(\lambda), \epsilon_2(\lambda), \epsilon_3(\lambda)$ are the maximum advantages over all attackers on Assumptions 1,2,3, respectively, then for any attacker $\mathcal{A}$ on our system we have the following:

Lemma	Result	Range
5.4	$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{VCI}} - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Res}} \leq 2\max(\epsilon_1, \epsilon_2)$	
5.5	$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Res}} - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{KG}_0} \leq \epsilon_1$	
5.6	$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{KG}_{i-1}} - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{KG}_i} \leq L\epsilon_2$	$1 \leq i \leq L-1$
5.7	$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{KG}_L} - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{SICI}} \leq L\epsilon_3$	
5.8	$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{KG}_i'} - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{KG}_{i-1}'} \leq L\epsilon_2$	$1 \leq i \leq L-1$
5.9	$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{KG}_0'} - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{ICI}} \leq L\epsilon_1$	

Recall that $\mathsf{SICI} = \mathsf{KG}_{L-1}'$, by adding all the inequalities:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{VCI}} - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{ICI}} \leq 2\max(\epsilon_1, \epsilon_2) + \epsilon_1 + 2L(L-1)\epsilon_2 + L(\epsilon_1 + \epsilon_3).$$

If the premise of the theorem is true, ϵ_1, ϵ_2, ϵ_3, are negligible functions of λ. Since L is a polynomial of λ, we conclude that the advantage of all PPT attackers is negligible. □

We remark that one may define $\texttt{Encap}^*$ differently for less game transitions, as long as it still achieves smoothness.

6. FUTURE DIRECTIONS

IMPROVE LEAKAGE FRACTION. A promising direction is to improve the leakage allowed from each secret key as the fraction of its size. It seems that our results can be generalized by using multiple tags in the secret key, but the security analysis is more complicated.

MULTIPLE-KEY LEAKAGE. In all leakage-resilient IBE systems, including the ones presented in this paper, leakage is allowed from only one secret key per identity. Although, this can be easily achieved by generating the randomness of the secret-key algorithm using a pseudo-random generator, leakage from multiple keys might be useful in HIBE (hierarchical identity-based encryption) and ABE (attribute-based encryption) systems, based on IBE constructions. In these cases different secret keys have to be generated for the same identities, and as a result it is more difficult to apply leakage-resilient techniques.

MASTER SECRET KEY LEAKAGE. As we saw no leakage is allowed from the master secret key of our systems. We assumed that it is totally hidden from the adversary. It is an interesting open question if there exist IBE systems resilient to master secret key leakage. Since there is a generic transformation of any IBE system to a signature scheme having as signing key the master secret key of the IBE system, it will provide constructions of leakage-resilient signature schemes.

LEAKAGE-RESILIENT HIBE. Finally, it is interesting to see whether leakage-resilient HIBE systems exist, or how existing techniques (like tagging) can be applied. to this setting.

Acknowledgement

Yevgeniy Dodis is supported by NSF grants CNS-0831299 and CNS-0716690. Brent Waters is supported by NSF CNS-0716199, CNS-0915361, and CNS-0952692, Air Force Office of Scientific Research (AFO SR) under the MURI award for "Collaborative policies and assured information sharing" (Project PRESIDIO), Department of Homeland Security Grant 2006-CS-001-000001-02 (subaward 641), and the Alfred P. Sloan Foundation.

7. REFERENCES

[1] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous Hardcore Bits and Cryptography against Memory Attacks. In *TCC*, pages 474–495, 2009.

[2] Joël Alwen, Yevgeniy Dodis, Moni Naor, Gil Segev, Shabsi Walfish, and Daniel Wichs. Public-Key Encryption in the Bounded-Retrieval Model. In *EUROCRYPT*, pages 113–134, 2010.

[3] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-Resilient Public-Key Cryptography in the Bounded-Retrieval Model. In *CRYPTO*, pages 36–54, 2009.

[4] Dan Boneh and Xavier Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *EUROCRYPT*, pages 223–238, 2004.

[5] Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. In *CRYPTO*, pages 213–229, 2001.

[6] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-Efficient Identity Based Encryption Without Pairings. In *FOCS*, pages 647–657, 2007.

[7] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-Resilient Functions and All-or-Nothing Transforms. In *EUROCRYPT*, pages 343–360, 2000.

[8] Ran Canetti, Shai Halevi, and Jonathan Katz. A Forward-Secure Public-Key Encryption Scheme. In *EUROCRYPT*, pages 255–271, 2003.

[9] Sherman S. M. Chow, Yevgeniy Dodis, Yannis Rouselakis, and Brent Waters. Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions, 2010. Full version.

[10] Giovanni Di Crescenzo, Richard J. Lipton, and Shabsi Walfish. Perfectly Secure Password Protocols in the Bounded Retrieval Model. In *TCC*, pages 225–244, 2006.

[11] Yevgeniy Dodis, Kristiyan Haralambiev, Adriana Lopez-Alt, and Daniel Wichs. Efficient Public-Key Cryptography in the Presence of Key Leakage. Cryptology ePrint Archive, Report 2010/154, 2010.

[12] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On Cryptography with Auxiliary Input. In *STOC*, pages 621–630, 2009.

[13] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[14] Yevgeniy Dodis and Krzysztof Pietrzak. Leakage-Resilient Pseudorandom Functions and Side-Channel Attacks on Feistel Networks. In *CRYPTO*, 2010. To appear.

[15] Stefan Dziembowski. Intrusion-Resilience Via the Bounded-Storage Model. In *TCC*, pages 207–224, 2006.

[16] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-Resilient Secret Sharing. In *FOCS*, pages 227–237, 2007.

[17] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-Resilient Cryptography. In *FOCS*, pages 293–302, 2008.

[18] Sebastian Faust, Eike Kiltz, Krzysztof Pietrzak, and Guy Rothblum. Leakage-Resilient Signatures. In *TCC*, pages 455–479, 2010.

[19] Sebastian Faust, Tal Rabin, Leonid Reyzin, Eran Tromer, and Vinod Vaikuntanathan. Protecting Circuits from Leakage: the Computationally-Bounded and Noisy Cases. In *EUROCRYPT*, pages 135–156, 2010.

[20] Craig Gentry. Practical Identity-Based Encryption Without Random Oracles. In *EUROCRYPT*, pages 445–464, 2006.

[21] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. In *STOC*, pages 197–206, 2008.

[22] Shafi Goldwasser and Silvio Micali. Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information. In *STOC*, pages 365–377, 1982.

[23] J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten. Lest We Remember: Cold Boot Attacks on Encryption Keys. In *USENIX Security Symposium*, pages 45–60, 2008.

[24] Yuval Ishai, Amit Sahai, and David Wagner. Private Circuits: Securing Hardware against Probing Attacks. In *CRYPTO*, pages 463–481, 2003.

[25] Jonathan Katz and Vinod Vaikuntanathan. Signature Schemes with Bounded Leakage Resilience. In *ASIACRYPT*, pages 703–720, 2009.

[26] Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *CRYPTO*, pages 104–113, 1996.

[27] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *CRYPTO*, pages 388–397, 1999.

[28] Allison B. Lewko and Brent Waters. New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In *TCC*, pages 455–479, 2010.

[29] Silvio Micali and Leonid Reyzin. Physically Observable Cryptography (Extended Abstract). In *TCC*, pages 278–296, 2004.

[30] Moni Naor and Gil Segev. Public-Key Cryptosystems Resilient to Key Leakage. In *CRYPTO*, pages 18–35, 2009.

[31] Noam Nisan. Extracting Randomness: How and Why - A survey. In *IEEE Conference on Computational Complexity*, pages 44–58, 1996.

[32] Krzysztof Pietrzak. A Leakage-Resilient Mode of Operation. In *EUROCRYPT*, pages 462–482, 2009.

[33] Ronald L. Rivest. All-or-Nothing Encryption and the Package Transform. In *FSE*, pages 210–218, 1997.

[34] Adi Shamir. Identity-Based Cryptosystems and Signature Schemes. In *CRYPTO*, pages 47–53, 1984.

[35] Brent Waters. Efficient Identity-Based Encryption Without Random Oracles. In *EUROCRYPT*, pages 114–127, 2005.