Anonymous Constant-Size Ciphertext HIBE From Asymmetric Pairings

Somindu C. Ramanna and Palash Sarkar

Applied Statistics Unit
Indian Statistical Institute
203, B.T. Road, Kolkata
India 700108.
e-mail: {somindu_r,palash}@isical.ac.in

Abstract. We present a new hierarchical identity based encryption (HIBE) scheme with constant-size ciphertexts that can be implemented using the most efficient bilinear pairings, namely, Type-3 pairings. In addition to being fully secure, our scheme is anonymous. The HIBE is obtained by extending an asymmetric pairing based IBE scheme due to Lewko and Waters. The extension uses the approach of Boneh-Boyen-Goh to obtain constant-size ciphertexts and that of Boyen-Waters for anonymity. Security argument is based on the dual-system technique of Waters. The resulting HIBE is the only known scheme using Type-3 pairings achieving constant-size ciphertext, security against adaptive-identity attacks and anonymity under static assumptions without random oracles.
Keywords: identity-based encryption(IBE), constant-size ciphertext hierarchical IBE, asymmetric pairings, dual-system encryption

1 Introduction

The notion of identity-based encryption (IBE) was introduced by Shamir [18] and the first IBE schemes appeared later [6, 3]. In IBE, a sender encrypts a message using the receiver's identity itself as the public key and a central authority called private key generator (PKG) generates and securely distributes decryption keys corresponding to identities of different users. Hierarchical IBE (HIBE), proposed by [11, 12], reduces the workload of the PKG by allowing it to delegate the key generation ability to lower-level entities. As a result, an individual user can conveniently obtain a decryption key from a lower-level entity instead of obtaining it from the PKG.

Type-3 Pairings: Most practical (H)IBE schemes are built using a bilinear pairing which maps $\mathbb{G}_1 \times \mathbb{G}_2$ to $\mathbb{G}_T$, where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups of the same order. Well-known examples of such maps arise by suitably choosing $\mathbb{G}_1$ and $\mathbb{G}_2$ to be groups of elliptic curve points and $\mathbb{G}_T$ to be a subgroup of the multiplicative group of a finite field. From an implementation point of view, it is most efficient to use bilinear maps where the (common) group order is prime and it is computationally difficult to find an isomorphism from $\mathbb{G}_1$ to $\mathbb{G}_2$ or vice versa. Such pairings are called Type-3 pairings [5, 19, 10]. Less efficient alternatives are when $\mathbb{G}_1$ and $\mathbb{G}_2$ are same (called Type-1 pairings) or when the common group order is a composite number (called composite-order pairings). IBE or HIBE schemes based on Type-3 pairings would have the fastest algorithms and the most compact representations of parameters.

Constant-Size Ciphertext HIBE: In HIBE, identities consist of tuples of varying lengths. Encryption of a message is done for a specified identity tuple. In many HIBE schemes, as the length of the identity tuple increases, so does the length of the resulting ciphertext. Consequently, the bandwidth requirement for communicating the ciphertext also increases.

The solution to this issue is to require the ciphertext size to be independent of the length of the identity tuple. Then, irrespective of the length of the identity tuple, the bandwidth required for the ciphertext would be the same. Such a scheme is called a constant-size ciphertext HIBE. The first such HIBE scheme was proposed by Boneh, Boyen and Goh [2]. While the scheme itself is quite elegant, its proof of security was in a very restricted attack model, the so-called selective-identity model. Lewko and Waters [15] provided the first constant-size ciphertext HIBE scheme which is secure against the usual adaptive-identity attacks. The drawback, however, was that the scheme in [15] used pairings on composite order groups and could not be instantiated with the more efficient Type-3 pairings.

In the following, we use the abbreviation CC-HIBE to denote HIBE schemes with constant-size ciphertexts. We clarify that the constant size here only refers to the number of group elements in the ciphertext. The size of representation of the group elements, however, needs to increase if the value of the security parameter increases.

Anonymity: In (H)IBE schemes with anonymity, ciphertexts do not reveal any information about the identity of the recipient. Abdalla *et.al.* [1] first formalised the notion of anonymity and used it to construct public key encryption with keyword search (PEKS). PEKS enables search on encrypted documents based on some keywords and this capability for search is delegateable. Anonymous HIBE schemes provide means to extend PEKS to more sophisticated primitives such as public key encryption with temporary keyword search (PETKS) and identity-based encryption with keyword search (IBEKS). The first construction of anonymous HIBE without random oracles was given by [4] with security in the selective-id model. Later constructions by [17, 7] could achieve security in the adaptive-id setting but were based on composite-order pairings. Two other constructions [8, 16] used asymmetric pairings but with security in the selective-id model.

1.1 Our Contributions

Our main motivation in this work is to obtain a constant-size ciphertext HIBE which can be implemented using Type-3 pairings. This allows the benefits of having constant-size ciphertexts to be combined with the efficiency benefits of using Type-3 pairings. These efficiency considerations are attained while retaining the usual provable guarantees, namely security against adaptive-identity attacks, use of static hardness assumptions, no degradation of security with increase in the depth of the HIBE and the avoidance of random oracles.

The provable properties are achieved using the extremely useful idea of dual-system encryption introduced by Waters [20]. This technique was used by Lewko and Waters [15] to construct an IBE and a CC-HIBE scheme based on composite-order pairings. The authors in [15] went on to convert their composite-order pairing based IBE scheme to one which can be instantiated using Type-3 pairings. However, no such conversion was done for the HIBE scheme in [15] and the authors do not make any remark on whether this can be done or how difficult it would be to do so.

The starting point of our work are the IBE schemes in [15]. Two IBE schemes are given in [15] where the first one is in the setting of composite order groups and the second one is in the Type-3 setting. The IBE in the composite order setting is not anonymous (shown in [7]) due to the following reason – the identity-hash in both the ciphertext and key live in the same subgroup; moreover, elements used to create the hash are public thus providing a test for the recipient identity for any ciphertext. On the other hand, the Type-3 variant, which we refer to as "LW-IBE", is anonymous. This is because ciphertexts live in $\mathbb{G}_1$, keys in $\mathbb{G}_2$ and the elements required to create the hash in $\mathbb{G}_2$ are kept secret. Hence there would be no way to test whether a given ciphertext is encrypted to a particular identity or not. However, there has been no proof of anonymity in any follow-up work. The first contribution of the current work is to show that the LW-IBE is anonymous. Two static (though non-standard) computational assumptions (which we denote as LW1 and LW2) along with decision bilinear Diffie-Hellman (DBDH) assumption are used in [15] to show the security of LW-IBE. For proving anonymity, we need to introduce a new computational assumption, called A1, which is again static, but, non-standard.

The second contribution of this paper is to extend the LW-IBE to a constant-size ciphertext HIBE. At a very basic level, the idea for obtaining constant-size ciphertexts is to use the identity hashing technique suggested in [2] over existing IBE schemes. We will refer to this as BBG-hash or BBG-extension. We do not take the path of converting the composite-order pairing based HIBE of [15]. Techniques for such conversions have been proposed by Freeman [9] and Lewko [14]. The latter uses *dual pairing vector spaces* (DPVSs) constructed over pairing groups to simulate features of composite order pairings. But it seems hard to retain the constant size of ciphertexts using these conversion techniques. Instead, we start with LW-IBE and extend it to a CC-HIBE by plugging in the BBG-hash. One complication in doing so arises. In the dual-system technique, two kinds of ciphertexts and keys are defined – *normal* and *semi-functional*. Semi-functional components are required only for proving secuirty and are generated using some secret elements during simulation. The main elements of a dual system proof would be appropriately defining semi-functional components and generating them using a problem instance in the reduction ensuring correct distribution of all elements provided to the attacker. Extending the decryption key of LW-IBE to the decryption key of a HIBE in a straightforward manner does not retain the structure required for a dual-system proof. Our way of tackling this is to add additional components to the decryption key. On the face of it, this complicates the key generation and delegation mechanisms. However, somewhat counter-intuitively, adding this extra level of complication allows the security reductions to go through.

An offshoot of the extension is that the scheme becomes anonymous. This is because in LW-IBE, the semi-functional space (for both ciphertexts and keys) is created using some secret elements (part of the master secret). The same elements are implicitly used in creating ciphertexts and keys. In case of a direct extension to HIBE, all these elements may have to be revealed in the public parameters to facilitate re-randomisation during delegation of keys. This makes the scheme non-anonymous but at the same time affects dual system arguments for which keeping the elements secret is essential. The way out is to make the scheme anonymous. We also provide a proof of anonymity based on a static assumption.

The computational assumptions required to obtain CPA-security are those used in [15] along with the new assumption required to show that the LW-IBE is anonymous. The last assumption is used to prove the anonymity of the HIBE scheme.

2 Preliminaries

Some basic notation, definitions and the complexity assumptions used in our proofs are presented in this section.

2.1 Notation

For a set $\mathcal{X}$, the notation $x_1, \ldots, x_k \in_{\mathrm{R}} \mathcal{X}$ (or $x_1, \ldots, x_k \xleftarrow{\mathrm{R}} \mathcal{X}$) indicates that $x_1, \ldots, x_k$ are elements of $\mathcal{X}$ chosen independently at random according to some distribution R. We use two notations interchangeably. The uniform distribution is denoted by U. For a (probabilistic) algorithm $\mathcal{A}$, $x \longleftarrow \mathcal{A}(\cdot)$ means that x is chosen according to the output distribution of $\mathcal{A}$ (which of course may be determined by its input). For two integers $a < b$, the notation $[a, b]$ represents the set $\{x \in \mathbb{Z} : a \leq x \leq b\}$. Let $\mathbb{G}$ be a finite cyclic group and $\mathbb{G}^{\times}$ denote the set of generators of $\mathbb{G}$. Fix a generator $P_1 \in \mathbb{G}^{\times}$. The discrete logarithm of an element $Q \in \mathbb{G}$ to base P_1 is written as $\mathsf{dlog}_{P_1} Q$.

2.2 Bilinear pairings

A bilinear pairing is given by a 7-tuple $\mathcal{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ where $\mathbb{G}_1 = \langle P_1 \rangle$, $\mathbb{G}_2 = \langle P_2 \rangle$ are written additively and $\mathbb{G}_T$, a multiplicatively written group, all having the same order p and $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a map with the following properties.

1. *Bilinear:* For $P_1, Q_1 \in \mathbb{G}_1$ and $P_2, Q_2 \in \mathbb{G}_2$, the following holds: $e(P_1, P_2 + Q_2) = e(P_1, P_2)e(P_1, Q_2)$ and $e(P_1 + Q_1, P_2) = e(P_1, P_2)e(Q_1, P_2)$.
2. *Non-degenerate:* If $e(P_1, P_2) = 1_T$, the identity element of $\mathbb{G}_T$, then either P_1 is the identity of $\mathbb{G}_1$ or P_2 is the identity of $\mathbb{G}_2$.
3. *Efficiently computable:* The function e should be efficiently computable.

Three main types of pairings have been identified in the literature [19, 10].

Type-1 In this type, the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ are the same.
Type-2 $\mathbb{G}_1 \neq \mathbb{G}_2$ and an efficiently computable isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ is known.
Type-3 Here, $\mathbb{G}_1 \neq \mathbb{G}_2$ and no efficiently computable isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$ are known.

It has been reported [5, 19, 10] that from an implementation point of view, Type-3 pairings are the fastest to compute and further provide the most compact description of group elements. So, building functionalities which can be instantiated with such pairings is of practical interest. This work is entirely based on Type-3 pairings. The terms 'Type-3 pairing' and 'asymmetric pairing' are used interchangeably in the rest of the paper.
Note: We introduce some notation: fix $P_1 \in \mathbb{G}_1^{\times}$ and $P_2 \in \mathbb{G}_2^{\times}$; for elements $R_1 \in \mathbb{G}_1$ and $R_2 \in \mathbb{G}_2$, the notation $R_1 \sim R_2$ indicates that $\mathsf{dlog}_{P_1} R_1 = \mathsf{dlog}_{P_2} R_2$. The fixed generators P_1 and P_2 will be clear from the context.

2.3 Complexity Assumptions

Here, we define certain hardness assumptions in Type-3 setting that are needed for the security reductions. In all the assumptions stated below, $\mathcal{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ is an asymmetric pairing and $\mathscr{A}$ is a probabilistic polynomial time (PPT) algorithm $\mathscr{A}$ that outputs 0 or 1.

Assumption LW1 [15]. Define a distribution $\mathcal{D}$ as follows: $F_1 \xleftarrow{\text{U}} \mathbb{G}_1^{\times}$; $F_2 \xleftarrow{\text{U}} \mathbb{G}_2^{\times}$, $a, b, s \xleftarrow{\text{U}} \mathbb{Z}_p$, $Y_1 \xleftarrow{\text{U}} \mathbb{G}_1$;

$$\mathcal{D} = (\mathcal{G}, F_1, bsF_1, sF_1, aF_1, ab^2F_1, bF_1, b^2F_1, asF_1, b^2sF_1, b^3F_1, b^3sF_1, F_2, bF_2).$$

The advantage of $\mathscr{A}$ in solving the LW1 problem is given by

$$\mathsf{Adv}_{\mathcal{G}}^{\text{LW1}}(\mathscr{A}) = |\Pr[\mathscr{A}(\mathcal{D}, ab^2sF_1) = 1] - \Pr[\mathscr{A}(\mathcal{D}, Y_1) = 1]|.$$

The (ε, t)-LW1 assumption holds in $\mathcal{G}$ if for any adversary $\mathscr{A}$ running in time at most t, $\mathsf{Adv}_{\mathcal{G}}^{\text{LW1}}(\mathscr{A}) \leq \varepsilon$.

Assumption LW2 [15]. Let distribution $\mathcal{D}$ be defined as follows: $F_1 \xleftarrow{\text{U}} \mathbb{G}_1^{\times}$; $F_2 \xleftarrow{\text{U}} \mathbb{G}_2^{\times}$, $d, b, c, x \xleftarrow{\text{U}} \mathbb{Z}_p$, $Y_2 \xleftarrow{\text{U}} \mathbb{G}_2$;

$$\mathcal{D} = (\mathcal{G}, F_1, dF_1, d^2F_1, bxF_1, dbxF_1, d^2xF_1, F_2, dF_2, bF_2, cF_2).$$

$\mathscr{A}$'s advantage in solving the LW2 problem is given by

$$\mathsf{Adv}_{\mathcal{G}}^{\text{LW2}}(\mathscr{A}) = |\Pr[\mathscr{A}(\mathcal{D}, bcF_2) = 1] - \Pr[\mathscr{A}(\mathcal{D}, Y_2) = 1]|.$$

The (ε, t)-LW2 assumption is that, for any t-time algorithm $\mathscr{A}$, $\mathsf{Adv}_{\mathcal{G}}^{\text{LW2}}(\mathscr{A}) \leq \varepsilon$.

Decisional Bilinear Diffie-Hellman in Type-3 pairings (DBDH-3) [5]. Let $F_1 \xleftarrow{\text{U}} \mathbb{G}_1^{\times}$, $F_2 \xleftarrow{\text{U}} \mathbb{G}_2^{\times}$, $x, y, z \xleftarrow{\text{U}} \mathbb{Z}_p$ and $Y_T \xleftarrow{\text{U}} \mathbb{G}_T$. Denote by $\mathcal{D}$, the distribution $(\mathcal{G}, F_1, xF_1, yF_1, zF_1, F_2, xF_2, yF_2)$. Define $\mathscr{A}$'s advantage in solving the DBDH-3 problem as follows.

$$\mathsf{Adv}_{\mathcal{G}}^{\text{DBDH-3}}(\mathscr{A}) = |\Pr[\mathscr{A}(\mathcal{D}, e(F_1, F_2)^{xyz}) = 1] - \Pr[\mathscr{A}(\mathcal{D}, Y_T) = 1]|.$$

We say that the (ε, t)-DBDH-3 assumption holds in $\mathcal{G}$ if $\mathsf{Adv}_{\mathcal{G}}^{\text{DBDH-3}}(\mathscr{A}) \leq \varepsilon$ for every algorithm $\mathscr{A}$ running in time at most t.

Assumption A1. Let $F_1 \xleftarrow{\text{U}} \mathbb{G}_1^{\times}$, $F_2 \xleftarrow{\text{U}} \mathbb{G}_2^{\times}$, $a, z, d, s, x \xleftarrow{\text{U}} \mathbb{Z}_p$ and

$$\mathcal{D} = (\mathcal{G}, F_1, zF_1, dzF_1, azF_1, adzF_1, szF_1, F_2, zF_2, aF_2, xF_2, (dz - ax)F_2).$$

The A1 problem is to decide, given $(\mathcal{D}, Y_1)$, whether $Y_1 = sdzF_1$ or $Y_1 \in_{\text{U}} \mathbb{G}_1$. The advantage of algorithm $\mathscr{A}$ in solving A1 is defined as

$$\mathsf{Adv}_{\mathcal{G}}^{\text{A1}}(\mathscr{A}) = |\Pr[\mathscr{A}(\mathcal{D}, sdzF_1) = 1] - \Pr[\mathscr{A}(\mathcal{D}, Y_1) = 1]|,$$

where $Y_1 \in_{\text{U}} \mathbb{G}_1$. The (ε, t)-A1 assumption is that any t-time algorithm $\mathscr{A}$ has $\mathsf{Adv}_{\mathcal{G}}^{\text{A1}}(\mathscr{A}) \leq \varepsilon$.

Discussion. We introduce assumption A1 to show anonymity of LW-IBE as well as our HIBE scheme. The challenge in A1 is an element $Z_1 \in \mathbb{G}_1$; the task is to decide whether $Z_1 = sdzF_1$ or random. Suppose we can successfully create $e(F_1, F_2)^{sdz\delta}$ (for some δ such that δF_2 is given in the instance) using elements in the instance, then the problem becomes easy to solve – just check for equality with $e(Z_1, \delta F_2)$. If they are equal then Z_1 is real; otherwise Z_1 is random. Since s and d appear in separate elements in $\mathbb{G}_1$, the only possible way is to compute $e(Z_1, zF_2)$ and compare it to $e((dz - ax)F_2, szF_1)$ after cancelling out $e(F_1, F_2)^{axsz}$. But this extra term cannot be cancelled since a and x appear in separate elements of $\mathbb{G}_2$. So our assumption is meaningful and there does not seem to be any way of efficiently solving A1.

Let DDH1 (resp. DDH2) be the decision Diffie-Hellman assumption in group $\mathbb{G}_1$ (resp. $\mathbb{G}_2$). It is well-known that in Type-3 setting these problems are computationally hard. The problem LW1 contains an embedded instance of DDH1. The elements sF_1 and ab^2F_1 are provided in the instance and it is required to determine whether Y_1 equals ab^2sF_1 or Y_1 is random. Similarly, LW2 contains an embedded instance of DDH2: the elements bF_2 and cF_2 are provided in the instance and it is required to determine whether Y_2 equals bcF_2 or Y_2 is random. As a result, an algorithm to solve DDH1 (resp. LW1) implies an algorithm to solve LW1 (resp. LW2) so that we can say that LW1 (resp. LW2) is no harder than DDH1 (resp. DDH2). The other direction, however, is not clear and it is due to this reason that the assumptions are considered non-standard.

Similar to the above, the problem A1 contains an embedded instance of DDH1. If $P_1 = zF_1$, $P_2 = zF_2$, then the elements $P_1, dP_1, sP_1, Z_1, P_2$ (present in the A1-instance) will form a proper DDH1 instance where it is required to determine whether $Z_1 = sdP_1 = sdzF_1$ or not. Hence a DDH1 solver can be used to solve A1. On the other hand, the converse is not known to hold.

2.4 Hierarchical Identity-Based Encryption

A HIBE scheme consists of five probabilistic polynomial time (in the security parameter) algorithms – Setup, $\mathsf{Encrypt}$, KeyGen, $\mathsf{Delegate}$ and $\mathsf{Decrypt}$.

- Setup: based on an input security parameter κ, generates and outputs the public parameters $\mathcal{PP}$ and the master secret $\mathcal{MSK}$.

– KeyGen: inputs an identity vector $\mathbf{id}$ and master secret $\mathcal{MSK}$ and outputs the secret key $\mathcal{SK}_{\mathbf{id}}$ corresponding to $\mathbf{id}$.
– $\mathsf{Encrypt}$: inputs an identity $\mathbf{id}$, a message M and returns a ciphertext $\mathcal{C}$.
– $\mathsf{Delegate}$: takes as input a depth ℓ identity vector $\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell)$, a secret key $\mathcal{SK}_{\mathbf{id}}$ and an identity $\mathsf{id}_{\ell+1}$; returns a secret key for the identity vector $(\mathsf{id}_1, \ldots, \mathsf{id}_{\ell+1})$.
– $\mathsf{Decrypt}$: inputs a ciphertext $\mathcal{C}$, an identity vector $\mathbf{id}$, secret key $\mathcal{SK}_{\mathbf{id}}$ and returns either the corresponding message M or $\perp$ indicating failure.

2.5 Anonymous CPA-Secure HIBE

The security game defined below captures both anonymity and security against a chosen plaintext attack for a HIBE scheme. This model, which we call $\mathsf{ano\text{-}ind\text{-}cpa}$, is equivalent to the standard security notions for CPA-security and anonymity and has been used earlier in [8, 7].
Setup: The challenger runs the Setup algorithm of the HIBE and gives the public parameters to $\mathscr{A}$.
Phase 1: $\mathscr{A}$ makes a number of key extraction queries adaptively. For a query on an identity vector $\mathbf{id}$, the challenger responds with a key $\mathcal{SK}_{\mathbf{id}}$.
Challenge: $\mathscr{A}$ provides two message-identity pairs $(M_0, \widehat{\mathbf{id}}_0)$ and $(M_1, \widehat{\mathbf{id}}_1)$ as challenge with the restriction that neither $\widehat{\mathbf{id}}_0, \widehat{\mathbf{id}}_1$ nor any of their prefixes should have been queried in **Phase 1**. The challenger then chooses a bit β uniformly at random from $\{0, 1\}$ and returns an encryption $\widehat{\mathcal{C}}$ of M_β under the identity $\widehat{\mathbf{id}}_\beta$ to $\mathscr{A}$.
Phase 2: $\mathscr{A}$ issues more key extraction queries as in **Phase 1** with the restriction that no queried identity $\mathbf{id}$ is a prefix of either $\widehat{\mathbf{id}}_0$ or $\widehat{\mathbf{id}}_1$.
Guess: $\mathscr{A}$ outputs a bit β'.

If $\beta = \beta'$, then $\mathscr{A}$ wins the game. The advantage of $\mathscr{A}$ in breaking the security of the HIBE scheme in the game $\mathsf{ano\text{-}ind\text{-}cpa}$ given by

$$\mathsf{Adv}_{\mathrm{HIBE}}^{\mathsf{ano\text{-}ind\text{-}cpa}}(\mathscr{A}) = \left|\Pr[\beta = \beta'] - \frac{1}{2}\right|.$$

The HIBE scheme is said to be (ε, t, q)-$\mathsf{ANO\text{-}IND\text{-}ID\text{-}CPA}$ secure if every t-time adversary making at most q queries has $\mathsf{Adv}_{\mathrm{HIBE}}^{\mathsf{ano\text{-}ind\text{-}cpa}}(\mathscr{A}) \leq \varepsilon$.

3 Lewko-Waters IBE

This section reviews the asymmetric pairing-based IBE construction of Lewko-Waters [15]. The description in [15] consists of the usual ciphertexts and keys as well as the so-called semi-functional ciphertexts and keys. We use a compact notation to denote normal and semi-functional ciphertexts and keys. The group elements shown in curly brackets { } are the semi-functional components. To get the scheme itself, these components should be ignored.

Let $\mathcal{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ be an asymmetric pairing. Pick $Q_1, U_1 \in_{\mathrm{U}} \mathbb{G}_1$ and $Q_2, U_2 \in \mathbb{G}_2$ be such that $Q_2 \sim Q_1$ and $U_2 \sim U_1$. Choose $F_2 \xleftarrow{\mathrm{U}} \mathbb{G}_2^{\times}$, $a, v, v' \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and define $V_2 = vF_2, V_2' = v'F_2$. Let $\tau = v + av'$ so that $\tau F_2 = V_2 + aV_2'$. Identities are elements of $\mathbb{Z}_p$. The public parameters and master secret are given by

$\mathcal{PP}$: $(P_1, aP_1, \tau P_1, Q_1, aQ_1, \tau Q_1, U_1, aU_1, \tau U_1, e(P_1, P_2)^\alpha)$
$\mathcal{MSK}$: $(\alpha P_2, P_2, V_2, V_2', Q_2, U_2, F_2)$.

The randomisers for the ciphertext and key are s and w, r_1, r_2 respectively. These are elements of $\mathbb{Z}_p$. For the semi-functional components, μ, σ and γ, π are chosen at random from $\mathbb{Z}_p$. Elements $V_1', F_1 \in \mathbb{G}_1$ are such that $V_1' \sim V_2'$ and $F_1 \sim F_2$.

Ciphertext:

$C_0 = M \cdot e(P_1, P_2)^{\alpha s}$
$C_{1,1} = s(\mathsf{id}Q_1 + U_1),\ C_{1,2} = as(\mathsf{id}Q_1 + U_1)\{+\mu\sigma F_1\},$
$C_{1,3} = -\tau s(\mathsf{id}Q_1 + U_1)\{-\mu\sigma V_1'\}$
$C_{2,1} = sP_1,\ C_{2,2} = asP_1\{+\mu F_1\},\ C_{2,3} = -\tau s P_1\{-\mu V_1'\}$

Key:

$K_{1,1} = wP_2 + r_1 V_2\{-a\gamma F_2\},\ K_{1,2} = r_1 V_2'\{+\gamma F_2\},\ K_{1,3} = r_1 F_2$
$K_{2,1} = \alpha P_2 + w(\mathsf{id}Q_2 + U_2) + r_1 V_2\{-a\gamma\pi F_2\},$
$K_{2,2} = r_2 V_2'\{+\gamma\pi F_2\},\ K_{2,3} = r_2 F_2$

Lewko and Waters show that this scheme is adaptively secure without random oracles under three non-standard but static assumptions – LW1, LW2 and DBDH-3. Since the elements Q_2, U_2 are in the master secret there seems to be no way to check whether a given ciphertext is encrypted to a particular identity or not. In other words, this scheme is anonymous. We provide a proof in the ANO-IND-ID-CPA model (described in Section 2.5) which encompasses both CPA-security and anonymity.

Theorem 1. *If the* $(\varepsilon_{\text{LW1}}, t')$-LW1*,* $(\varepsilon_{\text{LW2}}, t')$-LW2*,* $(\varepsilon_{\text{DBDH-3}}, t')$-DBDH-3 *and* $(\varepsilon_{\text{A1}}, t')$-A1 *assumptions hold, then* LW-IBE *is* (ε, t, q)-ANO-IND-ID-CPA *secure where*

$$\varepsilon \leq \varepsilon_{\text{LW1}} + q\varepsilon_{\text{LW2}} + \varepsilon_{\text{DBDH-3}} + \varepsilon_{\text{A1}}$$

and $t = t' - O(q\rho)$*, where* ρ *is an upper bound on the time required for one scalar multiplication in* $\mathbb{G}_1$ *or* $\mathbb{G}_2$*.*

Proof. Let $\mathscr{A}$ be any t-time adversary against LW-IBE in the ano-ind-cpa. The proof follows a hybrid argument over a sequence of $q+4$ games – $\text{Game}_{real}, \text{Game}_0, \text{Game}_1, \ldots, \text{Game}_q, \text{Game}_{M\text{-}rand}, \text{Game}_{final}$ – between $\mathscr{A}$ and a simulator $\mathscr{B}$, where the games are defined as follows.

- Game_{real}: the real security game ano-ind-cpa.
- Game_0: challenge ciphertext is semi-functional.
- Game_k $(1 \leq k \leq q)$: first k keys returned to the adversary are semi-functional and the rest are normal.
- $\text{Game}_{M\text{-}rand}$: the challenge ciphertext encrypts a random message under one of the challenge identities.
- Game_{final}: both message and challenge identity are random in the challenge ciphertext.

Let X_{real}, X_k, $X_{M\text{-}rand}$ and X_{final} denote the events that the adversary wins in Game_{real}, Game_k, $\text{Game}_{M\text{-}rand}$ and Game_{final} for $0 \leq k \leq q$ respectively. Note that, in Game_{final}, the challenge ciphertext is an encryption of a random message under a random identity vector. Hence β is statistically hidden from the adversary's view implying that $\Pr[X_{final}] = 1/2$. From [15], we know that $|\Pr[X_{real}] - \Pr[X_0]| \leq \varepsilon_{\text{LW1}}$, $|\Pr[X_{k-1}] - \Pr[X_k]| \leq \varepsilon_{\text{LW2}}$ and $|\Pr[X_q] - \Pr[X_{M\text{-}rand}]| \leq \varepsilon_{\text{DBDH-3}}$.

We now show that $\Pr[X_{M\text{-}rand}] - \Pr[X_{final}] \leq \varepsilon_{\text{A1}}$. Consider a simulator $\mathscr{B}$ playing the game ano-ind-cpa with $\mathscr{A}$. At this stage all keys are semi-functional and the message encrypted in the challenge ciphertext is random. Let $(\mathcal{G}, F_1, zF_1, dzF_1, azF_1, adzF_1, szF_1, F_2, zF_2, aF_2, xF_2, (dz - ax)F_2, Z_1)$ be the instance of A1 provided to $\mathscr{B}$. Let $Z_1 = c \cdot sdzF_1$. $\mathscr{B}$ has to determine whether $c = 1$ or $c \in_{\text{U}} \mathbb{Z}_p$. The game is simulated as follows.
Set-Up: Pick $\alpha, v, v', y, u \xleftarrow{\text{U}} \mathbb{Z}_p$ and set the parameters as

$P_1 = zF_1,\ V_2 = vF_2,\ V_2' = v'F_2,\ Q_1 = y(dzF_1),\ U_1 = u(dzF_1),$
$aP_1 = azP_1,\ aQ_1 = y(adzF_1),\ aU_1 = u(adzF_1),$

Similarly compute the elements τP_1, τQ_1 and τU_1. Compute $e(P_1, P_2)^\alpha = e(zF_1, zF_2)^\alpha$. $\mathscr{B}$ returns $\mathcal{PP}$ to $\mathscr{A}$. $\mathscr{B}$ knows $P_2 = zF_2$ and α but not Q_2 and U_2.

Key Extraction Phases 1 and 2: $\mathscr{B}$ picks $w, r_1, r_2 \xleftarrow{\text{U}} \mathbb{Z}_p$, $\gamma \xleftarrow{\text{U}} \mathbb{Z}_p^\times$ and $\pi' \xleftarrow{\text{U}} \mathbb{Z}_p$. It then computes the key for the k-th identity id_k as follows.

$$K_{1,1} = w(zF_2) + r_1 V_2 - \gamma a F_2,\ K_{1,2} = r_1 V_2' + \gamma F_2,\ K_{1,3} = r_1 F_2$$
$$K_{2,1} = \alpha z F_2 + w(y\mathsf{id}_k + u)(dz - ax)F_2 + r_2 V_2 - \gamma\pi'(aF_2),$$
$$K_{2,2} = r_2 V_2' + w(y\mathsf{id}_k + u)xF_2 + \gamma\pi' F_2,\ K_{2,3} = r_2 F_2,$$

setting $\pi = \pi' + \gamma^{-1} w(y\mathsf{id}_k + u)x$. Since $\gamma^{-1} w(y\mathsf{id}_k + u)x$ is additively randomised by π', π has the correct distribution in $\mathscr{A}$'s view. $\mathscr{B}$ returns $\mathcal{SK}_{\mathsf{id}_k} = ((K_{1,i}, K_{2,i})_{i=1,2,3})$ to $\mathscr{A}$. The following calculation shows that $K_{2,1}$ and $K_{2,2}$ are well-formed.

The following calculation shows that $K_{2,1}$ and $K_{2,2}$ are well-formed.

$$\begin{aligned}
K_{2,1} &= \alpha z F_2 + w(y\mathsf{id}_i + u)(dz - ax)F_2 + r_2 V_2 - \gamma\pi'(aF_2) \\
&= \alpha P_2 + w(y\mathsf{id}_i + u)(dz - ax)F_2 + r_2 V_2 - \gamma(\pi - \gamma^{-1} w(y\mathsf{id}_i + u)x)(aF_2) \\
&= \alpha P_2 + w(y\mathsf{id}_i + u)dzF_2 - w(y\mathsf{id}_i + u)axF_2 + r_2 V_2 - a\gamma\pi F_2 + w(y\mathsf{id}_i + u)(axF_2) \\
&= \alpha P_2 + w(\mathsf{id}_i Q_2 + U_2) + r_2 V_2 - a\gamma\pi F_2 \\
K_{2,2} &= r_2 V_2' + w(y\mathsf{id}_i + u)xF_2 + \gamma\pi' F_2 \\
&= r_2 V_2' + w(y\mathsf{id}_i + u)(xF_2) + \gamma(\pi - \gamma_1^{-1} w(y\mathsf{id}_i + u)x)F_2 \\
&= r_2 V_2' + w(y\mathsf{id}_i + u)(xF_2) + \gamma\pi F_2 - w(y\mathsf{id}_i + u)xF_2 \\
&= r_2 V_2' + \gamma\pi F_2.
\end{aligned}$$

Challenge: $\mathscr{B}$ receives two pairs of messages and identities $(M_0, \widehat{\mathsf{id}}_0)$ and $(M_1, \widehat{\mathsf{id}}_1)$ from $\mathscr{A}$. It chooses $\beta \xleftarrow{\text{U}} \{0,1\}$ and $a', \xi \xleftarrow{\text{U}} \mathbb{Z}_p$ at random and generates a semi-functional challenge ciphertext as follows.

$$C_0 \xleftarrow{\text{U}} \mathbb{G}_T$$
$$C_{1,1} = (y\widehat{\mathsf{id}}_\beta + u)Z_1,\ C_{1,2} = a'(y\widehat{\mathsf{id}}_\beta + u)Z_1 + \xi F_1,$$
$$C_{1,3} = -v(y\widehat{\mathsf{id}}_\beta + u))Z_1 - v'a'(y\widehat{\mathsf{id}}_\beta + u)Z_1 - v'\xi F_1,$$
$$C_{2,1} = szF_1,\ C_{2,2} = a'szF_1,\ C_{2,3} = -v(szF_1) - v'a'(szF_1),$$

where $a' = a + \mu'$, $\mu = \mu' sz$ and $\xi = \mu\sigma'$. The challenge ciphertext $\widehat{\mathcal{C}} = (C_0, C_{1,1}, C_{1,2}, C_{1,3}, C_{2,1}, C_{2,2}, C_{2,3})$ is returned to $\mathscr{A}$. The computations below illustrate that $\widehat{\mathcal{C}}$ is a semi-functional encryption with $\sigma = \sigma' + cd(y\widehat{\mathsf{id}}_\beta + u)$.

$$\begin{aligned}
C_{1,2} &= a'(y\widehat{\mathsf{id}}_\beta + u)Z_1 + \xi F_1 \\
&= (a + \mu')(y\widehat{\mathsf{id}}_\beta + u)csdzF_1 + \mu\sigma' F_1 \\
&= a(y\widehat{\mathsf{id}}_\beta + u)csdzF_1 + \mu' h(\widehat{\mathbf{id}}_\beta)csdzF_1 + \mu\sigma' F_1 \\
&= as(\widehat{\mathsf{id}}_\beta Q_1 + U_1) + (\mu' sz)(cd(y\widehat{\mathsf{id}}_\beta + u))F_1 + \mu\sigma' F_1 \\
&= as(\widehat{\mathsf{id}}_\beta Q_1 + U_1) + \mu(cd(y\widehat{\mathsf{id}}_\beta + u))F_1 + \mu\sigma' F_1 \\
&= as(\widehat{\mathsf{id}}_\beta Q_1 + U_1) + \mu\sigma F_1
\end{aligned}$$

Observe that $C_{1,1} = s(\widehat{\mathsf{id}}_\beta Q_1 + U_1) = (c \cdot (y\widehat{\mathsf{id}}_\beta + u))(sdzF_1)$. If $c = 1$, then $\sigma = \sigma' + d(y\widehat{\mathsf{id}}_\beta + u)$ and $\widehat{\mathcal{C}}$ is encrypted under $\widehat{\mathbf{id}}_\beta$. Otherwise, c is random, causing $(y\widehat{\mathsf{id}}_\beta + u)$ and consequently the target identity and σ to be random quantities.

Guess: $\mathscr{A}$ returns its guess β' of β.
If the algorithm $\mathscr{B}$ returns 1 when $\beta = \beta'$ and 0 otherwise, it can solve the A1 instance with advantage

$$\begin{aligned}\mathsf{Adv}^{\mathrm{A1}}_{\mathcal{G}}(\mathscr{B}) &= |\Pr[\beta = \beta' | Z_1 \text{ is real}] - \Pr[\beta = \beta' | Z_1 \text{ is random}]| \\ &= |\Pr[X_{M\text{-}rand}] - \Pr[X_{final}]|.\end{aligned}$$

Now, $\mathscr{A}$'s advantage in winning the game is given by

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}cpa}}_{\mathrm{LW-IBE}}(\mathscr{A}) &= \left|\Pr[X_{real}] - \frac{1}{2}\right| \\ &= |\Pr[X_{real}] - \Pr[X_{final}]| \\ &\leq |\Pr[X_{real}] - \Pr[X_0]| + \sum_{k=1}^{q} (|\Pr[X_{k-1}] - \Pr[X_k]|) \\ &\quad + |\Pr[X_{q,1}] - \Pr[X_{M\text{-}rand}]| + |\Pr[X_{M\text{-}rand}] - \Pr[X_{final}]| \\ &\leq \varepsilon_{\mathrm{LW1}} + q\varepsilon_{\mathrm{LW2}} + \varepsilon_{\mathrm{DBDH\text{-}3}} + \varepsilon_{\mathrm{A1}}\end{aligned}$$

□

4 Anonymous HIBE from LW-IBE

In this section, we present our HIBE scheme, $\mathcal{LW\text{-}AHIBE}$, resulting from a BBG-type extension of the LW IBE scheme. A straightforward BBG-type extension would lead to problems in adopting the dual system methodology. We introduce some new elements to overcome this problem. The construction is based on a Type-3 prime-order pairing with group order p. Identities are variable length tuples of elements from $\mathbb{Z}_p^{\times}$ with maximum length h.

The first step towards obtaining constant-size ciphertexts is to add elements $(Q_{1,j})_{j\in[1,h]}, U_1 \in \mathbb{G}_1$ to the public parameters. These are used to create the identity hash – for an identity $\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell)$, the hash is given by $\sum_{j=1}^{\ell} \mathsf{id}_j Q_{1,j} + U_1$. This replaces the hash in LW-IBE without affecting the number of elements in the ciphertext. To facilitate key extraction, the corresponding elements in $\mathbb{G}_2$ also are provided. We introduce some notation here: the tuple $(P_1, (Q_{1,j})_{j\in[1,h]}, U_1)$ is denoted $\mathbf{Q}_1$ and let its $\mathbb{G}_2$ counterpart be $\mathbf{Q}_2$. Also present in the master secret of LW-IBE are the elements V_2, V_2', F_2 that provide cancellation analogous to the composite order setting. In the HIBE setting, these elements along with $\mathbf{Q}_2$, must be made public to assist in re-randomisation during delegation. Once these are made public, nothing is kept secret except for α. This acts as a stumbling block against a dual system proof. In a proof within the dual system framework, some secret elements are needed to create the so-called semi-functional components that are central to this proof methodology. In the composite order setting, this is achieved by keeping one subgroup hidden from the attacker which essentially forms the semi-functional space. Similarly, schemes based on dual pairing vector spaces have some vectors in the dual bases hidden that assist in generating the semi-functional space. But the strategy for HIBE extension of LW-IBE chalked out above, requires everything to be made public (except α), which in turn limits our ability to define a semi-functional space.

Our solution to this problem is to keep $\mathbf{Q}_2$ in the master secret. In a way, some elements of the group $\mathbb{G}_2$ are hidden and provide the basis for generating semi-functional components. To support delegation, suitably randomised copies of the key components are provided in the key itself. This technique was introduced by Boyen and Waters [4] to construct an anonymous HIBE scheme. V_2, V_2', F_2 are public to help in re-randomisation during delegation; this ensures proper distribution of the delegated key. Note that $\mathbf{Q}_2$ contains precisely the elements required to check whether a ciphertext is encrypted to a particular identity or not.

A by-product of keeping this tuple secret is anonymity. Thus our scheme is secure in the ANO-IND-ID-CPA security model (refer to Section 2.5).

We now present the scheme $\mathcal{LW}\text{-}\mathcal{AHIBE}$. A discussion on the security of $\mathcal{LW}\text{-}\mathcal{AHIBE}$ can be found in Section 5.

Construction

$\mathsf{Setup}(\kappa)$: Let h denote the maximum depth of the HIBE. Choose random generators $P_1 \in \mathbb{G}_1$ and $P_2 \in \mathbb{G}_2$; elements $Q_{1,1}, \ldots, Q_{1,h}, U_1 \xleftarrow{\text{U}} \mathbb{G}_1$ and $Q_{2,1}, \ldots, Q_{2,h}, U_2 \in \mathbb{G}_2$ such that $Q_{2,j} \sim Q_{1,j}$ for all $1 \leq j \leq h$ and $U_2 \sim U_1$. Let $F_2 \in \mathbb{G}_2$ be chosen at random and v, v' be chosen randomly from $\mathbb{Z}_p$. Set $V_2 = vF_2$, $V_2' = v'F_2$. Pick α, a at random from $\mathbb{Z}_p$. Set $\tau = v + av'$ so that $\tau F_2 = V_2 + aV_2'$.

$$
\begin{aligned}
\mathcal{PP} &: (P_1, aP_1, \tau P_1, U_1, aU_1, \tau U_1, (Q_{1,j}, aQ_{1,j}, \tau Q_{1,j})_{j \in [1,h]},\\
&\quad V_2, V_2', F_2, e(P_1, P_2)^{\alpha}).\\
\mathcal{MSK} &: (\alpha P_2, P_2, Q_{2,1}, \ldots, Q_{2,h}, U_2).
\end{aligned}
$$

$\mathsf{Encrypt}(M, \mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell), \mathcal{PP})$: Choose $s \xleftarrow{\text{U}} \mathbb{Z}_p$. Let $\mathcal{H}_i(\mathbf{id}) = \mathsf{id}_1 Q_{i,1} + \cdots + \mathsf{id}_\ell Q_{i,\ell} + U_i$ for $i = 1, 2$. The ciphertext is given by $\mathcal{C} = (C_0, C_{1,1}, C_{1,2}, C_{1,3}, C_{2,1}, C_{2,2}, C_{2,3})$ where the elements are computed as follows.

$$
\begin{aligned}
&C_0 = M \times e(P_1, P_2)^{\alpha s},\\
&C_{1,1} = s\mathcal{H}_1(\mathbf{id}),\ C_{1,2} = as\mathcal{H}_1(\mathbf{id}),\ C_{1,3} = -\tau s\mathcal{H}_1(\mathbf{id})\\
&C_{2,1} = sP_1,\ C_{2,2} = asP_1,\ C_{2,3} = -\tau sP_1
\end{aligned}
$$

$\mathsf{KeyGen}(\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell), \mathcal{MSK}, \mathcal{PP})$: Choose $w_1, w_2, r_1, r_2, r_3, r_4, (z_{1,j}, z_{2,j})_{j \in [\ell+1,h]} \xleftarrow{\text{U}} \mathbb{Z}_p$. The key consists of $6(n - \ell + 2)$ group elements computed as follows.

$$
\begin{aligned}
&K_{1,1} = w_1 P_2 + r_1 V_2,\ K_{1,2} = r_1 V_2',\ K_{1,3} = r_1 F_2\\
&K_{2,1} = \alpha P_2 + w_1 \mathcal{H}_2(\mathbf{id}) + r_2 V_2,\ K_{2,2} = r_2 V_2',\ K_{2,3} = r_2 F_2\\
&D_{j,1} = w_1 Q_{2,j} + z_{1,j} V_2,\ D_{j,2} = z_{1,j} V_2',\ D_{j,3} = z_{1,j} F_2 \text{ for } \ell + 1 \leq j \leq h
\end{aligned}
$$

$$
\begin{aligned}
&J_{1,1} = w_2 P_2 + r_3 V_2,\ J_{1,2} = r_3 V_2',\ J_{1,3} = r_3 F_2\\
&J_{2,1} = w_2 \mathcal{H}_2(\mathbf{id}) + r_4 V_2,\ J_{2,2} = r_4 V_2',\ J_{2,3} = r_4 F_2\\
&E_{j,1} = w_2 Q_{2,j} + z_{2,j} V_2,\ E_{j,2} = z_{2,j} V_2',\ E_{j,3} = z_{2,j} F_2 \text{ for } \ell + 1 \leq j \leq h.
\end{aligned}
$$

The secret key for $\mathbf{id}$ is given by $\mathcal{SK}_{\mathbf{id}} = (\mathcal{S}_1, \mathcal{S}_2)$, where $\mathcal{S}_1 = (K_{1,i}, K_{2,i}, D_{j,i})_{j \in [\ell+1,h], i=1,2,3}$ and $\mathcal{S}_2 = (J_{1,i}, J_{2,i}, E_{j,i})_{j \in [\ell+1,h], i=1,2,3}$. Notice that $\mathcal{S}_2$-components are almost same as $\mathcal{S}_1$-components except that the secret α is not embedded in $\mathcal{S}_2$. The set $\mathcal{S}_2$ is exclusively used for re-randomisation.

$\mathsf{Delegate}(\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell), \mathcal{SK}_{\mathbf{id}}, \mathsf{id}_{\ell+1}, \mathcal{PP})$: Let $\mathbf{id} : \mathsf{id}_{\ell+1}$ denote the $\ell + 1$-length identity vector $(\mathsf{id}_1, \ldots, \mathsf{id}_\ell, \mathsf{id}_{ell+1})$ obtained by appending $\mathsf{id}_{\ell+1}$ to $\mathbf{id}$. Choose $r_1', r_2', r_3', r_4', (z_{1,j}', z_{2,j}')_{j \in [\ell+2,h]} \xleftarrow{\text{U}} \mathbb{Z}_p$ and $w_1', w_2' \xleftarrow{\text{U}} \mathbb{Z}_p^\times$. The components of the key for the identity $\mathbf{id} : \mathsf{id}_{\ell+1}$ are computed as follows.

$$
\begin{aligned}
&K_{1,1} \leftarrow K_{1,1} + w_1' J_{1,1} + r_1' V_2 && K_{2,1} \leftarrow K_{2,1} + \mathsf{id}_{\ell+1} D_{\ell+1,1} + w_1'(J_{2,1} + \mathsf{id}_{\ell+1} E_{\ell+1,1}) + r_2' V_2\\
&K_{1,2} \leftarrow K_{1,2} + w_1' J_{1,2} + r_1' V_2' && K_{2,2} \leftarrow K_{2,2} + \mathsf{id}_{\ell+1} D_{\ell+1,2} + w_1'(J_{2,2} + \mathsf{id}_{\ell+1} E_{\ell+1,2}) + r_2' V_2'\\
&K_{1,3} \leftarrow K_{1,3} + w_1' J_{1,3} + r_1' F_2 && K_{2,3} \leftarrow K_{2,3} + \mathsf{id}_{\ell+1} D_{\ell+1,3} + w_1'(J_{2,3} + \mathsf{id}_{\ell+1} E_{\ell+1,3}) + r_2' F_2
\end{aligned}
$$

$$
\begin{aligned}
&J_{1,1} \leftarrow w_2' J_{1,1} + r_3' V_2 && J_{2,1} \leftarrow w_2'(J_{2,1} + \mathsf{id}_{\ell+1} E_{\ell+1,1}) + r_4' V_2\\
&J_{1,2} \leftarrow w_2' J_{1,2} + r_3' V_2' && J_{2,2} \leftarrow w_2'(J_{2,2} + \mathsf{id}_{\ell+1} E_{\ell+1,2}) + r_4' V_2'\\
&J_{1,3} \leftarrow w_2' J_{1,3} + r_3' F_2 && J_{2,3} \leftarrow w_2'(J_{2,3} + \mathsf{id}_{\ell+1} E_{\ell+1,3}) + r_4' F_2
\end{aligned}
$$

For $j = \ell + 2, \ldots, h$,

$D_{j,1} \leftarrow D_{j,1} + w'_1 E_{j,1} + z'_{1,j} V_2 \quad D_{j,2} \leftarrow D_{j,2} + w'_1 E_{j,2} + z'_{1,j} V'_2 \quad D_{j,3} \leftarrow D_{j,3} + w'_1 E_{j,3} + z'_{1,j} F_2$

$E_{j,1} \leftarrow w'_2 E_{j,1} + z'_{2,j} V_2 \quad E_{j,2} \leftarrow w'_2 E_{j,2} + z'_{2,j} V'_2 \quad E_{j,3} \leftarrow w'_2 E_{j,3} + z'_{2,j} F_2$

The above procedure essentially re-randomises all components of the key. As a result the distribution of a key obtained using delegation is the same as the distribution of a key obtained using the key generation procedure. To note the re-randomisation consider the following change of scalars for the modified key.

$w_1 \leftarrow w_1 + w'_1 w_2$; $\quad w_2 \leftarrow w'_2 w_2$;

$r_1 \leftarrow r_1 + r'_1 + w'_1 r_3$; $\quad r_3 \leftarrow w'_2 r_3 + r'_3$;

$r_2 \leftarrow r_2 + r'_2 + \mathsf{id}_{\ell+1} z_{1,\ell+1} + w'_2(r_4 + \mathsf{id}_{\ell+1} z_{2,\ell+1})$; $\quad r_4 \leftarrow w'_2(r_4 + \mathsf{id}_{\ell+1} z_{2,\ell+1}) + r'_4$;

$z_{1,j} \leftarrow z_{1,j} + z'_{1,j} + w'_1 z_{2,j+1}$ for $j = \ell + 2, \ldots, h$ $\quad z_{2,j} \leftarrow w'_2 z_{2,j} + z'_{2,j}$ for $j = \ell + 2, \ldots, h$

These new randomisers are properly distributed by the choice of $w'_1, w'_2, r'_1, r'_2, r'_3, r'_4, (z'_{1,j}), (z'_{2,j})$.
$\mathsf{Decrypt}(\mathcal{C}, \mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell), \mathcal{SK}_{\mathbf{id}}, \mathcal{PP})$: Decryption is done as follows.

$$M = C_0 \times \frac{e(C_{1,1}, K_{1,1}) e(C_{1,2}, K_{1,2}) e(C_{1,3}, K_{1,3})}{e(C_{2,1}, K_{2,1}) e(C_{2,2}, K_{2,2}) e(C_{2,3}, K_{2,3})} \tag{1}$$

Correctness of decryption of the HIBE scheme follows directly from that of LW-IBE since the decryption procedure remains the same – the additional delegation components do not play any role in decryption. Observe that computing the ratio of pairings in Equation (1) using $J_{1,i}, J_{2,i}$ $(i = 1, 2, 3)$ instead of $K_{1,i}, K_{2,i}$ results in 1_T (the identity of $\mathbb{G}_T$).

5 Security of $\mathcal{LW}$-$\mathcal{AHIBE}$

We first provide some basic intuition underlying the proof with respect to different stages of security analysis (within the dual system framework), highlighting the similarities and differences with LW-IBE security proof. Then, a detailed security analysis of $\mathcal{LW}$-$\mathcal{AHIBE}$ is presented in Section 5.2.

5.1 Ideas Underlying the Security Proof

The first step is to define semi-functional (sf) ciphertexts and keys. The definition of sf-ciphertext remains the same as LW-IBE. The keys of $\mathcal{LW}$-$\mathcal{AHIBE}$ are significantly different from LW-IBE. We formulate the definition of sf-keys on the basis of the following observations.

- Sf-components for $(K_{1,i}, K_{2,i})_{i=1,2}$ are identical to LW-IBE since only these components participate in decryption.
- It is required to define sf-components for $(D_{j,1}, D_{j,2})_{j \in [\ell+1,h]}$ though they are only used during delegation to create the identity-hash. This is because they share the randomiser w_1 with $K_{1,i}, K_{2,i}$ and this randomiser comes from a problem instance in the reductions.
- Once sf-components are defined for $\mathcal{S}_1$, it is natural to ask: is it necessary to define sf-parts for $\mathcal{S}_2$? The answer is yes since otherwise the fourth reduction fails, where $P_2, U_2, (Q_{2,j})$ are masked by a quantity that forces the keys to be semi-functional. We have already seen this in the context of LW-IBE (see Theorem 1).

We would like to emphasise that the definition of semi-functional components (in both ciphertexts and keys), complexity assumptions and the reductions are all inter-linked. Changing the structure of sf-keys may determine the assumption required or affect simulation in some reduction. Also, for the reductions to go through, the sf-components may have to be defined in a particular way. The structure of sf-components we have is in a sense, optimal, subject to assumptions and simulations we provide.

An outline of the four main reductions in the augmented security proof (including anonymity) of LW-IBE is as follows.

First reduction: The goal of this reduction is to show that an attacker cannot distinguish between a normal ciphertext and an sf-ciphertext. It is achieved via a reduction from the LW1 problem. An LW1 instance is embedded in the challenge ciphertext attempting to exploit the adversary's ability to detect the change in order to solve the problem.

Second reduction: In this reduction, it is shown that if the adversary can decide whether the response to the k-th key extraction query is normal or semi-functional, then LW2 problem can be solved. The k-th key is constructed from an instance of LW2 problem in such a way that the key is normal if the instance is 'real' and semi-functional otherwise.

Third reduction: Here, the message that the challenge ciphertext encrypts, is changed to a random element of $\mathbb{G}_T$. It is shown that solving the DBDH-3 problem is no harder than distinguishing between an sf-encryption of the real message from an sf-encryption of a random element of $\mathbb{G}_T$.

Fourth reduction: Challenge ciphertext encrypts a random message under a random identity. The identity-hash is created using the challenge in an instance of A1 problem thus making it real or random according to the distribution of the challenge.

This strategy does not directly extend to the hierarchical setting. Several challenges/restrictions emerge as we try to prove security of $\mathcal{LW}$-$\mathcal{AHIBE}$.

The first and the third reductions for $\mathcal{LW}$-$\mathcal{AHIBE}$ are the closest to the corresponding reductions for LW-IBE appearing in [15]. In these reductions, the simulations of the public parameters; the ciphertext elements; and the components of the key which are present in LW-IBE; are exactly the same as for LW-IBE. The only technicality is to ensure that the extra components of the key can be properly simulated without changing the corresponding assumptions (LW1 for the first reduction and DBDH-3 for the third reduction).

The second reduction presents some technical novelty. We need to extend the dual-system technique to handle this reduction. In this reduction, it is shown that the adversary cannot decide whether the response to the k-th key extraction query is normal or semi-functional. Compared to the LW-IBE, the key has additional components which are required for delegation and re-randomisation; moreover, these have semi-functional parts. A new technique is required to handle these simulations.

Partial semi-functionality: Consider the second reduction where the k-th key is made semi-functional. LW-IBE reduction embeds a pairwise independent function in the k-th key as well as the challenge ciphertext to ensure independent distribution of the scalars involved in the respective sf-components. This function is determined by the parameters used to create the identity-hash. An attempt to use the same strategy for $\mathcal{LW}$-$\mathcal{AHIBE}$, however, causes a problem. The reason is that the identity-hash is now present in three places – challenge ciphertext, $\mathcal{S}_1$ and $\mathcal{S}_2$. In addition, all these have sf-components. One possible way to deal with this is to embed a 3-wise independent function i.e., a degree-2 polynomial in the identity. As result the one extra group element is required in $\mathcal{PP}$ as well as $\mathcal{MSK}$. Also, encryption and key generation would each require an extra scalar multiplication and a squaring in the underlying field. The other way to get around the problem is to use two separate instances to generate the two hashes in the key. We follow the latter approach since the efficiency of the scheme remains unaffected alhtough the degradation is increased by a factor of 2. The key is changed from normal to semi-functional in two steps – first make $\mathcal{S}_1$ semi-functional followed by $\mathcal{S}_2$. We call a key partial semi-functional if $\mathcal{S}_1$ is semi-functional and $\mathcal{S}_2$ is normal.

The second step of the dual-system technique changes the key in the k-th response from normal to semi-functional (without the adversary noticing this). In our case, this is done in two sub-steps – the first step changes from normal to partial semi-functional and the second step changes from partial semi-functional to semi-functional. This leads to a slight degradation in the security bound by a factor of 2.

The fourth reduction is to show anonymity of the HIBE scheme. This is almost the same as the reduction that we have provided to show the anonymity of the LW-IBE. The only difference is that the extra elements of the key have to properly simulated.

5.2 Detailed Proof

As is typical in the dual-system technique, we first describe semi-functional ciphertexts and keys. These are required only in the reductions and not in the actual scheme.

Semi-functional ciphertext: Let $C'_0, C'_{1,1}, C'_{1,2}, C'_{1,3}, C'_{2,1}, C'_{2,2}, C'_{2,3}$ be ciphertext elements normally generated by the $\mathsf{Encrypt}$ algorithm for message M and identity $\mathbf{id}$. Let V'_1, F_1 be elements of $\mathbb{G}_1$ such that $V'_1 \sim V'_2$ and $F_1 \sim F_2$. Choose $\mu, \sigma \in \mathbb{Z}_p$ at random. The semi-functional ciphertext generation algorithm will modify the normal ciphertext as: $C_0 = C'_0$, $C_{1,1} = C'_{1,1}$, $C_{2,1} = C'_{2,1}$ and

$$C_{1,2} = C'_{1,2} + \mu\sigma F_1,\ C_{1,3} = C'_{1,3} - \mu\sigma V'_1, C_{2,2} = C'_{2,2} + \mu F_1,\ C_{2,3} = C'_{2,3} - \mu V'_1.$$

Semi-functional key: Let $(\mathcal{S}_1, \mathcal{S}_2)$ be the secret key generated by the KeyGen algorithm for identity $\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell)$ with $\mathcal{S}_1 = (K_{1,i}, K_{2,i}, D_{j,i})_{j\in[\ell+1,h], i=1,2,3}$, $\mathcal{S}_2 = (J_{1,i}, J_{2,i}, E_{j,i})_{j\in[\ell+1,h], i=1,2,3}$. Let $\gamma_1, \pi, \gamma_2, \eta, (\pi_j, \eta_j)_{j\in[\ell+1,h]}$ be uniform random elements chosen from $\mathbb{Z}_p$. The semi-functional key generation algorithm will modify the normal key as:

$$\begin{array}{ll} K_{1,1} = K_{1,1} - a\gamma_1 F_2,\ K_{1,2} = K_{1,2} + \gamma_1 F_2, & J_{1,1} = J_{1,1} - a\gamma_2 F_2,\ J_{1,2} = J_{1,2} + \gamma_2 F_2, \\ K_{2,1} = K_{2,1} - a\gamma_1 \pi F_2,\ K_{2,2} = K_{2,2} + \gamma_1 \pi F_2, & J_{2,1} = J_{2,1} - a\gamma_2 \eta F_2,\ J_{2,2} = J_{2,2} + \gamma_2 \eta F_2, \end{array}$$

For $j = \ell + 1, \ldots, h$

$$\begin{array}{ll} D_{j,1} = D_{j,1} - a\gamma_1 \pi_j F_2,\ D_{j,2} = D_{j,2} + \gamma_1 \pi_j F_2, & E_{j,1} = E_{j,1} - a\gamma_2 \eta_j F_2,\ E_{j,2} = E_{j,2} + \gamma_2 \eta_j F_2. \end{array}$$

The rest of the components remain unchanged.

Partial semi-functional key: In a partial semi-functional key, $\mathcal{S}_2$ is normal and $\mathcal{S}_1$ is semi-functional.

Note that definitions are similar to [15] except for the delegation and re-randomisation components. Since decryption is not affected by these components of the key, all the requirements for semi-functional keys and ciphertexts are satisfied. A pair of semi-functional ciphertext and key is called *nominally semi-functional* if $\sigma = \pi$ (condition that makes decryption successful).

Structure of the Proof. We consider the security model defined in Section 2.5. The proof is organised as a hybrid over a sequence of $2q + 4$ games defined as follows.

- Game$_{real}$: $\mathsf{ano\text{-}ind\text{-}cpa}$ game defined in Section 2.5.
- Game$_{0,1}$: the challenge ciphertext is semi-functional and all the keys returned to the adversary are normal.
- Game$_{k,0}$ (for $1 \le k \le q$): k-th key is partial semi-functional, the first $k - 1$ keys are semi-functional; the rest of the keys are normal.
- Game$_{k,1}$ (for $1 \le k \le q$): similar to Game$_{k,0}$ except that the k-th key is (fully) semi-functional.
- Game$_{M\text{-}rand}$: all keys are semi-functional and the challenge ciphertext encrypts a random message to the challenge identity.
- Game$_{final}$: similar to Game$_{M\text{-}rand}$ except that the challenge ciphertext now encrypts to a random identity vector.

These games are ordered as Game_{real}, $\mathrm{Game}_{0,1}$, $\mathrm{Game}_{1,0}, \mathrm{Game}_{1,1}, \ldots, \mathrm{Game}_{q,0}, \mathrm{Game}_{q,1}$, $\mathrm{Game}_{M\text{-}rand}$, Game_{final} in our hybrid argument. Let $X_\Box$ be events that $\mathscr{A}$ wins in $\mathrm{Game}_\Box$.

For the proof it will be convenient to use the following short-hand: denote by $h(\mathbf{id})$ the sum $\sum_{j=1}^{\ell} y_j \mathsf{id}_j + u$ and by $g(\mathbf{id})$ the sum $\sum_{j=1}^{\ell} \lambda_j \mathsf{id}_j + \nu$, where $y_1, \ldots, y_n, u, \lambda_1, \ldots, \lambda_n, \nu$ are elements of $\mathbb{Z}_p$ to be chosen in the proofs.

Theorem 2. *If the* $(\varepsilon_{\mathrm{LW1}}, t')$-LW1*,* $(\varepsilon_{\mathrm{LW2}}, t')$-LW2*,* $(\varepsilon_{\mathrm{DBDH\text{-}3}}, t')$-DBDH-3 *and* $(\varepsilon_{\mathrm{A1}}, t')$-A1 *assumptions hold, then* $\mathcal{LW}\text{-}\mathcal{AHIBE}$ *is* (ε, t, q)-ANO-IND-ID-CPA *secure where*

$$\varepsilon \leq \varepsilon_{\mathrm{LW1}} + 2q\varepsilon_{\mathrm{LW2}} + \varepsilon_{\mathrm{DBDH\text{-}3}} + \varepsilon_{\mathrm{A1}}$$

and $t = t' - O(q\rho)$*, where* ρ *is an upper bound on the time required for one scalar multiplication in* $\mathbb{G}_1$ *or* $\mathbb{G}_2$*.*

Proof. For any t-time adversary $\mathscr{A}$ against $\mathcal{LW}\text{-}\mathcal{AHIBE}$ in the ano-ind-cpa, its advantage in winning the game is given by

$$\mathsf{Adv}^{\text{ano-ind-cpa}}_{\mathcal{LW}\text{-}\mathcal{AHIBE}}(\mathscr{A}) = \left|\Pr[X_{real}] - \frac{1}{2}\right|.$$

We know that $\Pr[X_{final}] = \frac{1}{2}$ and hence we have

$$\begin{aligned}
\mathsf{Adv}^{\text{ano-ind-cpa}}_{\mathcal{LW}\text{-}\mathcal{AHIBE}}(\mathscr{A}) &= |\Pr[X_{real}] - \Pr[X_{final}]| \\
&\leq |\Pr[X_{real}] - \Pr[X_0]| + \sum_{k=1}^{q} (|\Pr[X_{k-1,1}] - \Pr[X_{k,0}]|) + \sum_{k=1}^{q} (|\Pr[X_{k,0}] - \Pr[X_{k,1}]|) \\
&\quad + |\Pr[X_{q,1}] - \Pr[X_{M\text{-}rand}]| + |\Pr[X_{M\text{-}rand}] - \Pr[X_{final}]| \\
&\leq \varepsilon_{\mathrm{LW1}} + 2q\varepsilon_{\mathrm{LW2}} + \varepsilon_{\mathrm{DBDH\text{-}3}} + \varepsilon_{\mathrm{A1}}
\end{aligned}$$

The last inequality follows from the lemmas 1, 2, 3, 4 and 5. In all the lemmas, $\mathscr{A}$ is a t-time adversary against $\mathcal{LW}\text{-}\mathcal{AHIBE}$ and $\mathscr{B}$ is an algorithm running in time t' that interacts with $\mathscr{A}$ and solves one of the three problems LW1, LW2, DBDH-3 or A1. $\Box$

Lemma 1. $|\Pr[X_{real}] - \Pr[X_{0,1}]| \leq \varepsilon_{\mathrm{LW1}}$.

Proof. The algorithm $\mathscr{B}$ receives the following instance of LW1

$$(F_1, bsF_1, sF_1, aF_1, ab^2F_1, bF_1, b^2F_1, asF_1, b^2sF_1, b^3F_1, b^3sF_1, F_2, bF_2, Z_1).$$

$\mathscr{B}$ has to determine whether $Z_1 = ab^2sF_1$ or $Z_1 \in_{\mathrm{U}} \mathbb{G}_1$. We will call Z_1 "real" in the former case and "random" otherwise. $\mathscr{B}$ simulates the security game as described below.

Set-Up: $\mathscr{B}$ chooses $\alpha, y, v', (y_j, \lambda_j)_{j\in[1,h]}, u, \nu \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and sets the parameters.

$$P_1 = b^2F_1 + yF_1, Q_{1,j} = \lambda_j(b^2F_1) + y_jF_1 \text{ for } 1 \leq j \leq h, U_1 = \nu(b^2F_1) + uF_1$$

$$V_2 = bF_2, V_2' = v'F_2.$$

This implicitly sets $P_2 = (b^2 + y)F_2$, $v = b$ and $\tau = b + av'$. Compute $aP_1 = ab^2F_1 + y(aF_1)$ and $\tau P_1 = b^3F_1 + v'(ab^2F_1) + y(bF_1) + yv'(aF_1)$. The elements $(aQ_{1,j}, \tau Q_{1,j})_{j\in[1,h]}, aU_1, \tau U_1$ are constructed similarly. Set $e(P_1, P_2)^\alpha = (e(b^3F_1 + y(bF_1), bF_2)e(P_1, yF_2))^\alpha$. The simulator gives the following public parameters to $\mathscr{A}$.

$$\mathcal{PP} = (P_1, Q_{1,1}, \ldots, Q_{1,h}, U_1, aP_1, aQ_{1,1}, \ldots, aQ_{1,h}, aU_1, \tau P_1, \tau Q_{1,1}, \ldots, \tau Q_{1,h}, \tau U_1, e(P_1, P_2)^\alpha).$$

Phases 1 and 2: $\mathscr{A}$ makes a number of key extract queries. $\mathscr{B}$ does not know $P_2, Q_{2,j}, U_2$ which are part of the master secret. The secret key for a query on **id** is constructed as follows. $\mathscr{B}$ chooses $r_1', r_2', r_3', r_4', (z_{1,j}', z_{2,j}')_{j\in[\ell+1,h]}, w_1, w_2 \in \mathbb{Z}_p$ at random and computes

$$
\begin{aligned}
&K_{1,1} = w_1 y F_2 + r_1'(bF_2),\ K_{1,3} = r_1' F_2 - w(bF_2),\ K_{1,2} = v' K_{1,3},,\\
&K_{2,1} = \alpha y F_2 + r_2'(bF_2) + wh(\mathbf{id})F_2,\ K_{2,3} = r_2' F_2 - (w_1 g(\mathbf{id}) + \alpha)(bF_2),\ K_{2,2} = v' K_{2,3},,\\
&D_{j,1} = w_1 y_j F_2 + z_{1,j}'(bF_2),\ D_{j,3} = z_{1,j}' F_2 - w_1 \lambda_j (bF_2),\ D_{j,2} = v' D_{j,3} \text{ for } \ell+1 \le j \le h,
\end{aligned}
$$

$$
\begin{aligned}
&J_{1,1} = w_2 y F_2 + r_3'(bF_2),\ J_{1,3} = r_3' F_2 - w_2(bF_2),\ J_{1,2} = v' J_{1,3},\\
&J_{2,1} = r_4'(bF_2) + w_2 h(\mathbf{id})F_2,\ J_{2,3} = r_4' F_2 - w_2 g(\mathbf{id})(bF_2),\ J_{2,2} = v' J_{2,3},\\
&E_{j,1} = w_2 y_j F_2 + z_{2,j}'(bF_2),\ E_{j,3} = z_{2,j}' F_2 - w_2 \lambda_j (bF_2),\ E_{j,2} = v' E_{j,3} \text{ for } \ell+1 \le j \le h,
\end{aligned}
$$

implicitly setting

$$
\begin{aligned}
&r_1 = r_1' - w_1 b,\ r_2 = r_2' - (w_1 g(\mathbf{id}) + \alpha)b,\\
&r_3 = r_3' - w_2 b,\ r_4 = r_4' - w_2 g(\mathbf{id})b,\\
&z_{1,j} = z_{1,j}' - w_1 \lambda_j b,\ z_{2,j} = z_{2,j}' - w_2 \lambda_j b \text{ for } j = \ell+1, \ldots, h.
\end{aligned}
$$

The following computation shows that the components are well-formed.

$$
\begin{aligned}
K_{1,1} &= wyF_2 + r_1'(bF_2) \\
&= wyF_2 + (r_1 + wb)bF_2 \\
&= w(yF_2 + b^2F_2) + r_1(bF_2) \\
&= wP_2 + r_1 V_2
\end{aligned}
\qquad
\begin{aligned}
K_{2,1} &= \alpha y F_2 + r_2'(bF_2) + wh(\mathbf{id})F_2 \\
&= \alpha y F_2 + (r_2 + (wg(\mathbf{id}) + \alpha)b)bF_2 + wh(\mathbf{id})F_2 \\
&= \alpha(yF_2 + b^2F_2) + r_2(bF_2) + w(g(\mathbf{id})b^2F_2 + h(\mathbf{id})F_2) \\
&= \alpha P_2 + w\mathcal{H}_2(\mathbf{id}) + r_2 V_2
\end{aligned}
$$

$$
\begin{aligned}
D_{1,j} &= wy_jF_2 + z_j'(bF_2) \\
&= wy_jF_2 + (z_{1,j} + w\lambda_j b)(bF_2) \\
&= w(y_jF_2 + \lambda_j b^2 F_2) + z_{1,j}(bF_2) \\
&= wQ_{2,j} + z_{1,j} V_2
\end{aligned}
$$

Following the same logic, it can be verified that $J_{1,1}, J_{2,1}, E_{j,1}$ are well-formed. Remaining components clearly have the right form.

Challenge: $\mathscr{B}$ receives two pairs $(M_0, \widehat{\mathbf{id}}_0)$ and $(M_1, \widehat{\mathbf{id}}_1)$ from $\mathscr{A}$. It chooses $\beta \in \{0,1\}$ at random. $\mathscr{B}$ computes the ciphertext for M_β under $\widehat{\mathbf{id}}_\beta$ as follows.

$$
\begin{aligned}
&C_0 = M_\beta \cdot \left(e(b^3sF_1 + y(bsF_1), bF_2)e(b^2sF_1 + y(sF_1), yF_2)\right)^\alpha = M_\beta \cdot e(P_1, P_2)^{\alpha s}\\
&C_{1,1} = g(\widehat{\mathbf{id}}_\beta)(b^2sF_1) + h(\widehat{\mathbf{id}}_\beta)(sF_1),\ C_{1,2} = g(\widehat{\mathbf{id}}_\beta)Z_1 + h(\widehat{\mathbf{id}}_\beta)(asF_1)\\
&C_{1,3} = -g(\widehat{\mathbf{id}}_\beta)(b^3sF_1) - h(\widehat{\mathbf{id}}_\beta)(bsF_1) - v'g(\widehat{\mathbf{id}}_\beta)Z_1 - v'h(\widehat{\mathbf{id}}_\beta)(asF_1)\\
&C_{2,1} = b^2sF_1 + y(sF_1),\ C_{2,2} = Z_1 + y(asF_1)\\
&C_{2,3} = -b^3sF_1 - y(bsF_1) - v'Z_1 - v'(asF_1).
\end{aligned}
$$

$\mathscr{B}$ returns $\widehat{\mathcal{C}} = (C_0, C_{1,1}, C_{1,2}, C_{1,3}, C_{2,1}, C_{2,2}, C_{2,3})$ to $\mathscr{A}$.

If $Z_1 = ab^2sF_1$, then it is easy to see that $\widehat{\mathcal{C}}$ is normal. Otherwise, $Z_1 = (ab^2s + \mu)F_1$ for some $\mu \in_{\mathrm{U}} \mathbb{Z}_p$ and $\widehat{\mathcal{C}}$ is semi-functional with $\mu = \mu$ and $\sigma = g(\widehat{\mathbf{id}}_\beta)$. The calculation below shows that $C_{1,2}$ is a properly formed (semi-functional) component.

$$\begin{aligned} C_{1,2} &= g(\widehat{\mathbf{id}}_\beta)Z_1 + h(\widehat{\mathbf{id}}_\beta)(asF_1) \\ &= g(\widehat{\mathbf{id}}_\beta)(ab^2s+\mu)F_1 + h(\widehat{\mathbf{id}}_\beta)(asF_1) \\ &= as(g(\widehat{\mathbf{id}}_\beta)b^2F_1 + h(\widehat{\mathbf{id}}_\beta)F_1) + \mu g(\widehat{\mathbf{id}}_\beta)F_1 \\ &= as\mathcal{H}_1(\widehat{\mathbf{id}}_\beta) + \mu\sigma F_1 \end{aligned}$$

Verification of the well-formedness of $C_{1,3}$, $C_{2,2}$ and $C_{2,3}$ follows the same pattern. Scalars $(\lambda_j)_{j\in[1,h]}, \nu$ are information theoretically hidden from $\mathscr{A}$'s view and hence $\sigma = g(\mathbf{id})$ appears to be uniformly and independently distributed with respect to all other information provided to $\mathscr{A}$.

Note that, to check whether $\widehat{\mathcal{C}}$ is semi-functional or not, $\mathscr{B}$ itself could try to decrypt it with a semi-functional key for $\widehat{\mathbf{id}}_\beta$. Any such attempt will fail due to the following reason – aF_2 is unavailable to $\mathscr{B}$; it could try to cancel out $-a\gamma F_2$ in $K_{1,1}$ or γF_2 in $K_{1,2}$ with some other elements; but we do not see how to achieve this keeping the link between $K_{1,1}$ and $K_{1,2}$ (via γ) intact, without knowing aF_2.

Guess: The adversary returns its guess β' to $\mathscr{B}$.

If Z_1 is real, $\widehat{\mathcal{C}}$ is normal and hence $\mathscr{B}$ simulates Game$_{real}$. Otherwise, Z_1 is random and $\widehat{\mathcal{C}}$ is semi-functional in which case, $\mathscr{B}$ simulates Game$_{0,1}$. Suppose that $\mathscr{B}$ returns 1 if $\beta = \beta'$ and 0 otherwise. Then it can solve the LW1 problem with advantage

$$\mathsf{Adv}_{\mathcal{G}}^{\mathrm{LW1}}(\mathscr{B}) = |\Pr[\beta = \beta'|Z_1 \text{ is real}] - \Pr[\beta = \beta'|Z_1 \text{ is random}]| = |\Pr[X_{real}] - \Pr[X_{0,1}]| \le \varepsilon_{\mathrm{LW1}}.$$

□

Lemma 2. $|\Pr[X_{k-1,1}] - \Pr[X_{k,0}]| \le \varepsilon_{\mathrm{LW2}}$ *for* $1 \le k \le q$.

Proof. Let $(F_1, dF_1, d^2F_1, bxF_1, dbxF_1, d^2xF_1, F_2, dF_2, bF_2, cF_2, Z_2)$ be the instance of LW2 that $\mathscr{B}$ receives. Let $Z_2 = (bc+\gamma)F_2$. $\mathscr{B}$'s task is to decide whether $\gamma = 0$ (Z_2 is real) or $\gamma \in_{\mathrm{U}} \mathbb{Z}_p$ (Z_2 is random).

Set-Up: $\mathscr{B}$ chooses $\alpha, a, y_v, y_1, \ldots, y_h, u, \lambda_1, \ldots, \lambda_h, \nu \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and computes parameters as follows. $P_1 = dF_1$, $Q_{1,j} = \lambda_j(dF_1) + y_jF_1$ for $1 \le j \le h$, $U_1 = \nu(dF_1) + uF_1$, $V_2 = -a(bF_2) + dF_2 + y_vF_2$ and $V_2' = bF_2$ setting $v = -ab + d + y_v$, $v' = b$ and $\tau = d + y_v$. The element τP_1 can be computed as $\tau P_1 = d^2F_1 + y_v(dF_1)$. The parameters $\tau Q_{1,j}$ for $1 \le j \le h$ and τU_1 are given by $\tau Q_{1,j} = \lambda_j(d^2F_1) + y_j(dF_1) + y_v\lambda_j(dF_1) + y_vy_jF_1$ and $\tau U_1 = \nu(d^2F_1) + u(dF_1) + y_v\nu(dF_1) + y_vuF_1$. The remaining parameters required to provide $\mathcal{PP}$ to $\mathscr{A}$ are computed using a, α and elements of the problem instance. Elements of the master secret key can also be obtained from the instance and randomisers chosen at setup.

Phases 1 and 2: The key extraction queries for identities $\mathbf{id}_1, \ldots, \mathbf{id}_q$ are answered in the following way. If $i < k$, a semi-functional key is returned and if $i > k$ a normal key is returned. $\mathscr{B}$ creates semi-functional keys using the master secret, a and F_2.

For $i = k$, $\mathscr{B}$ computes of $\mathcal{S}_1$ using the problem instance in the following manner. Let $\mathbf{id}_k = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell)$. $\mathscr{B}$ chooses $w_1', r_2', z_{1,\ell+1}', \ldots, z_{1,h}' \xleftarrow{\mathrm{U}} \mathbb{Z}_p$.

$$\begin{aligned} &K_{1,1} = w_1'P_2 - aZ_2 + y_v(cF_2),\ K_{1,2} = Z_2,\ K_{1,3} = cF_2 \\ &K_{2,1} = \alpha P_2 + w_1'(g(\mathbf{id}_k)(dF_2) + h(\mathbf{id}_k)F_2) + r_2'V_2 - ag(\mathbf{id}_k)Z_2 + y_vg(\mathbf{id}_k)(cF_2) - h(\mathbf{id}_k)cF_2 \\ &K_{2,2} = r_2'V_2' + g(\mathbf{id}_k)Z_2,\ K_{2,3} = r_2'F_2 + g(\mathbf{id}_k)(cF_2) \end{aligned}$$

and for $j = \ell+1, \ldots, h$, set

$$\begin{aligned} &D_{j,1} = w_1'Q_{2,j} + z_{1,j}'V_2 - y_j(cF_2) - a\lambda_jZ_2 + y_v\lambda_j(cF_2) \\ &D_{j,2} = z_{1,j}'V_2' + \lambda_jZ_2,\ D_{j,3} = z_{1,j}'F_2 + \lambda_j(cF_2) \end{aligned}$$

thus implicitly setting $w_1 = w'_1 - c$, $r_1 = c$, $r_2 = r'_2 + g(\mathbf{id}_k)c$ and $z_{1,j} = z'_{1,j} + \lambda_j c$ for $\ell + 1 \le j \le h$.

Let $\mathcal{S}_1 = (K_{1,i}, K_{2,i}, D_{j,i})_{j\in[\ell+1,h],i=1,2,3}$. The second set $\mathcal{S}_2 = (J_{1,i}, J_{2,i}, E_{j,i})_{j\in[\ell+1,h],i=1,2,3}$ is created normally. $\mathscr{B}$ returns $\mathcal{SK}_{\mathbf{id}_k} = (\mathcal{S}_1, \mathcal{S}_2)$ as the key for $\mathbf{id}_k$. If $Z_2 = bcF_2$ then the key for $\mathbf{id}_k$ is normal.

We show that $K_{2,1}$ is well-formed. Verifying the remaining parts can be done analogously.

$$\begin{aligned}
K_{2,1} &= \alpha P_2 + w'_1(g(\mathbf{id}_k)(dF_2) + h(\mathbf{id}_k)F_2) + r'_2 V_2 - ag(\mathbf{id}_k)Z_2 + y_v g(\mathbf{id}_k)(cF_2) - h(\mathbf{id}_k)cF_2 \\
&= \alpha P_2 + (w_1 + c)\mathcal{H}_2(\mathbf{id}_k) + (r_2 - g(\mathbf{id}_k)c)(-a(bF_2) + dF_2 + y_v F_2) \\
&\qquad + g(\mathbf{id}_k)(-abcF_2 + y_v cF_2) - h(\mathbf{id}_k)cF_2 \\
&= \alpha P_2 + (w_1 + c)\mathcal{H}_2(\mathbf{id}_k) + r_2 V_2 - g(\mathbf{id}_k)cdF_2 + g(\mathbf{id}_k)(abcF_2 - cy_v F_2) \\
&\qquad - g(\mathbf{id}_k)(abcF_2 - y_v cF_2) - h(\mathbf{id}_k)cF_2 \\
&= \alpha P_2 + (w_1 + c)\mathcal{H}_2(\mathbf{id}_k) + r_2 V_2 - g(\mathbf{id}_k)cdF_2 - h(\mathbf{id}_k)cF_2 \\
&= \alpha P_2 + (w_1 + c)\mathcal{H}_2(\mathbf{id}_k) + r_2 V_2 - c\mathcal{H}_2(\mathbf{id}_k) \\
&= \alpha P_2 + w_1\mathcal{H}_2(\mathbf{id}_k) + r_2 V_2.
\end{aligned}$$

If $Z_2 = (bc + \gamma)F_2$ the key will be partial semi-functional with $\gamma_1 = \gamma$, $\pi = g(\mathbf{id}_k)$ and $\pi_j = \lambda_j$ for $\ell + 1 \le j \le h$. It is straightforward to check that $\mathcal{SK}_{\mathbf{id}_k}$ is a properly formed partial sf-key. Also, since $(\lambda_j)_{j\in[1,h]}, \nu$ are information theoretically hidden from the adversary, $\pi, (\pi_j)_{j\in[\ell+1,h]}$ are uniformly and independently distributed in $\mathscr{A}$'s view.

$\mathscr{B}$ could attempt checking whether $\mathcal{SK}_{\mathbf{id}_k}$ is semi-functional by creating a sf-ciphertext for $\mathbf{id}_k$. Since $V'_1 = bF_1$ is not available to $\mathscr{B}$, the only way of doing this will lead to σ being the same as π (challenge ciphertext is created via this method). The ciphertext-key pair will be nominally semi-functional and thus provides no information to $\mathscr{B}$.

Challenge: $\mathscr{A}$ provides two message-identity pairs, $(M_0, \widehat{\mathbf{id}}_0)$ and $(M_1, \widehat{\mathbf{id}}_1)$ to $\mathscr{B}$. It chooses $\beta \in_{\mathrm{U}} \{0, 1\}$, generates the challenge ciphertext as shown below.

$$\begin{aligned}
&C_0 = M_\beta \cdot e(dbxF_1, dF_2)^\alpha \\
&C_{1,1} = g(\widehat{\mathbf{id}}_\beta)(dbxF_1) + h(\widehat{\mathbf{id}}_\beta)(bxF_1) \\
&C_{1,2} = ag(\widehat{\mathbf{id}}_\beta)(dbxF_1) + ah(\widehat{\mathbf{id}}_\beta)(bxF_1) - g(\widehat{\mathbf{id}}_\beta)(d^2xF_1) \\
&C_{1,3} = -y_v g(\widehat{\mathbf{id}}_\beta)(dbxF_1) - h(\widehat{\mathbf{id}}_\beta)(dbxF_1) - y_v h(\widehat{\mathbf{id}}_\beta)(bxF_1) \\
&C_{2,1} = dbxF_1,\ C_{2,2} = a(dbxF_1) - d^2xF_1,\ C_{2,3} = -y_v(dbxF_1).
\end{aligned}$$

This sets $s = bx$, $\mu = -d^2x$ and $\sigma = g(\widehat{\mathbf{id}}_\beta)$. Since $\lambda_1, \ldots, \lambda_h$ and ν are chosen uniformly at random from $\mathbb{Z}_p$, $\lambda_1 X_1 + \cdots + \lambda_h X_h + \nu$ is a pairwise independent function for variables $X_1, \ldots, X_h$ over $\mathbb{Z}_p$. As a result, $\pi = \lambda_1 \mathsf{id}_1 + \cdots + \lambda_\ell \mathsf{id}_\ell + \nu$ and $\sigma = \lambda_1 \widehat{\mathsf{id}}_1 + \cdots + \lambda_{\widehat{\ell}} \widehat{\mathsf{id}}_{\widehat{\ell}} + \nu$ are independent and uniformly distributed. $\mathscr{B}$ returns $\widehat{\mathcal{C}} = (C_{1,i}, C_{2,i})_{i=1,2,3}$.

To show that $\widehat{\mathcal{C}}$ is indeed distributed properly, we show that $C_{1,3}$ is well-formed. Along the same lines, one can check the well-formedness of $C_{1,2}$, $C_{2,2}$ and $C_{2,3}$.

$$\begin{aligned}
C_{1,3} &= -y_v g(\widehat{\mathbf{id}}_\beta)(dbxF_1) - h(\widehat{\mathbf{id}}_\beta)(dbxF_1) - y_v h(\widehat{\mathbf{id}}_\beta)(bxF_1) \\
&= -y_v g(\widehat{\mathbf{id}}_\beta)(dbxF_1) - y_v h(\widehat{\mathbf{id}}_\beta)(bxF_1) - h(\widehat{\mathbf{id}}_\beta)(dbxF_1) - g(\widehat{\mathbf{id}}_\beta)d^2bxF_1 + g(\widehat{\mathbf{id}}_\beta)d^2bxF_1 \\
&= -y_v bx\mathcal{H}_1(\widehat{\mathbf{id}}_\beta) - dbx\mathcal{H}_1(\widehat{\mathbf{id}}_\beta) + g(\widehat{\mathbf{id}}_\beta)d^2x(bF_1) \\
&= -\tau\mathcal{H}_1(\widehat{\mathbf{id}}_\beta) + \sigma\mu V'_1
\end{aligned}$$

Guess: $\mathscr{A}$ returns a bit β' as its guess for β.

When the instance is real, $\mathscr{B}$ simulates $\text{Game}_{k-1,1}$ and otherwise simulates $\text{Game}_{k,0}$. $\mathscr{B}$ returns 1 if $\mathscr{A}$ wins the game i.e., $\beta = \beta'$; otherwise it returns 0. Hence, $\mathscr{B}$ can solve the LW2 instance with advantage

$$\mathsf{Adv}_{\mathcal{G}}^{\text{LW2}}(\mathscr{B}) = |\Pr[\beta = \beta' | Z_2 \text{ is real}] - \Pr[\beta = \beta' | Z_2 \text{ is random}]| = |\Pr[X_{k-1,1}] - \Pr[X_{k,0}]|.$$

from which the statement of the lemma follows. □

Lemma 3. $|\Pr[X_{k,0}] - \Pr[X_{k,1}]| \leq \varepsilon_{\text{LW2}}$ *for* $1 \leq k \leq q$.

The proof is reminiscent of Lemma 2. The reason is as follows: the structure of $\mathcal{S}_2$ is identical to $\mathcal{S}_1$ if the αP_2 term is removed from $K_{2,1}$. Moreover, the simulator chooses α and creates αP_2 independent of the instance. Hence the simulation will be similar except that the instance is now embedded in $\mathcal{S}_2$ and $\mathcal{S}_1$ is made semi-functional independent of the instance.

Lemma 4. $|\Pr[X_{q,1}] - \Pr[X_{M\text{-}rand}]| \leq \varepsilon_{\text{DBDH-3}}$.

Proof. $\mathscr{B}$ receives $(F_1, aF_1, bF_1, sF_1, F_2, aF_2, bF_2, sF_2, Z_T)$ as an instance of the DBDH-3 problem where $Z_T = e(F_1, F_2)^{abs}$ (real) or $Z_T \in_{\text{U}} \mathbb{G}_T$ (random).
Set-Up: With $y, v, v', y_1, \ldots, y_h, u$ chosen at random from $\mathbb{Z}_p$, $\mathscr{B}$ sets the parameters as

$$P_1 = yF_1, P_2 = yF_2, aP_1 = y(aF_1), V_2 = vF_2, V_2' = v'F_2,\ \tau P_1 = yvF_1 + yv'(aF_1)$$

$$Q_{1,j} = y_j P_1 = y_j y F_1 \text{ for } 1 \leq j \leq h, U_1 = uP_1 = uyF_1, e(P_1, P_2)^{\alpha} = e(aF_1, bF_2)^{y^2}$$

implicitly setting $\alpha = ab$ and $\tau = v + av'$. The remaining parameters can be computed easily. $\mathscr{B}$ returns $\mathcal{PP}$ to $\mathscr{A}$.
Phases 1 and 2: When $\mathscr{A}$ asks for the secret key for the i'th identity $\mathbf{id}_i = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell)$, $\mathscr{B}$ chooses at random $w_1, w_2, r_1, r_2, r_3, r_4, (z_{1,j}, z_{2,j})_{j=1}^{h}$ and $\gamma_1', \gamma_1, \gamma_2, (\pi_j)_{j=1}^{h}, \eta, (\eta_j)_{j=1}^{h}$ from $\mathbb{Z}_p$ and computes a semi-functional key for $\mathbf{id}_i$ as follows.

$$\begin{aligned}
&K_{1,1} = w_1 P_2 + r_1 V_2 - \gamma_1(aF_2),\ K_{1,2} = r_1 V_2' + \gamma_1 F_2,\ K_{1,3} = r_1 F_2\\
&K_{2,1} = \gamma_1'(aF_2) + w_1 h(\mathbf{id}_i)(P_2) + r_2 V_2,\ K_{2,2} = r_2 V_2' + y(bF_2) - \gamma_1' F_2,\ K_{2,3} = r_2 F_2,\\
&D_{j,1} = w_1 Q_{2,j} + z_{1,j} V_2 - \gamma_1 \pi_j(aF_2),\ D_{j,2} = z_{1,j} V_2' + \gamma_1 \pi_j F_2,\ D_{j,3} = z_{1,j} F_2 \text{ for } \ell + 1 \leq j \leq h.
\end{aligned}$$

$$\begin{aligned}
&J_{1,1} = w_2 P_2 + r_3 V_2 - \gamma_2(aF_2),\ J_{1,2} = r_3 V_2' + \gamma_2 F_2,\ J_{1,3} = r_3 F_2\\
&J_{2,1} = w_2 h(\mathbf{id}_i)(P_2) + r_4 V_2 - \gamma_2 \eta(aF_2),\ J_{2,2} = r_4 V_2' + \gamma_2 \eta F_2,\ J_{2,3} = r_4 F_2,\\
&E_{j,1} = w_2 Q_{2,j} + z_{2,j} V_2 - \gamma_2 \eta_j(aF_2),\ E_{j,2} = z_{2,j} V_2' + \gamma \eta_j F_2,\ E_{j,3} = z_{2,j} F_2 \text{ for } \ell + 1 \leq j \leq h.
\end{aligned}$$

Here the relation a$\gamma_1' = by - \gamma_1 \pi$ is implicitly set by the simulator. Calculations provided below justify that $K_{2,1}$ and $K_{2,2}$ have the correct distribution. Other elements have the correct form and distribution.

$$\begin{aligned}
K_{2,1} &= \gamma_1'(aF_2) + w_1 h(\mathbf{id}_i)(P_2) + r_2 V_2\\
&= (by - \gamma_1 \pi)(aF_2) + w_1 h(\mathbf{id}_i)(P_2) + r_2 V_2\\
&= ab(yF_2) + w_1 h(\mathbf{id}_i)(P_2) + r_2 V_2 - a\gamma_1 \pi F_2\\
&= \alpha P_2 + w_1 h(\mathbf{id}_i)(P_2) + r_2 V_2 - a\gamma_1 \pi F_2.
\end{aligned}
\qquad
\begin{aligned}
K_{2,2} &= r_2 V_2' + y(bF_2) - \gamma_1' F_2\\
&= r_2 V_2' + y(bF_2) - (by - \gamma_1 \pi) F_2\\
&= r_2 V_2' + y(bF_2) - byF_2 + \gamma_1 \pi F_2\\
&= r_2 V_2' + \gamma_1 \pi F_2.
\end{aligned}$$

Observe that $\mathscr{B}$ does not know α or αF_2 and hence cannot create a normal key.
Challenge: $\mathscr{B}$ receives two pairs $(M_0, \widehat{\mathbf{id}}_0)$ and $(M_1, \widehat{\mathbf{id}}_1)$ from $\mathscr{A}$. It samples $\beta \xleftarrow{\text{U}} \{0, 1\}$, $\mu' \xleftarrow{\text{U}} \mathbb{Z}_p$ and generates a semi-functional challenge ciphertext as follows.

$$\begin{aligned}
&C_0 = M_\beta \times Z_T\\
&C_{1,1} = yh(\widehat{\mathbf{id}}_\beta) sF_1,\ C_{1,2} = h(\widehat{\mathbf{id}}_\beta)\mu' F_1,\ C_{1,3} = -vyh(\widehat{\mathbf{id}}_\beta)(sF_1) - v'h(\widehat{\mathbf{id}}_\beta)\mu' F_1\\
&C_{2,1} = y(sF_1),\ C_{2,2} = \mu' F_1,\ C_{2,3} = -yv(sF_1) - v'\mu' F_1
\end{aligned}$$

with $\mu' = asy + \mu$ and $\sigma = h(\widehat{\mathbf{id}}_\beta)$. The challenge ciphertext $\widehat{\mathcal{C}} = (C_0, C_{1,1}, C_{1,2}, C_{1,3}, C_{2,1}, C_{2,2}, C_{2,3})$ is returned to $\mathscr{A}$.

Guess: $\mathscr{A}$ returns its guess β' of β.

It is clear that $\widehat{\mathcal{C}}$ is a semi-functional encryption of M_β when $Z_T = e(P_1, P_2)^{abs}$. And when $Z_T \in_{\mathrm{U}} \mathbb{G}_T$ $\widehat{\mathcal{C}}$ would be a semi-functional encryption of a random message. Hence $\mathscr{B}$ simulates $\mathrm{Game}_{q,1}$ or Game_{final} according to Z_T being real or random respectively. If the algorithm $\mathscr{B}$ returns 1 when $\beta = \beta'$ and 0 otherwise, it can solve the DBDH-3 instance with advantage

$$\mathsf{Adv}_{\mathcal{G}}^{\text{DBDH-3}}(\mathscr{B}) = |\Pr[\beta = \beta' | Z_T \text{ is real}] - \Pr[\beta = \beta' | Z_T \text{ is random}]| = |\Pr[X_{q,1}] - \Pr[X_{final}]|.$$

□

Lemma 5. $|\Pr[X_{M\text{-}rand}] - \Pr[X_{final}]| \leq \varepsilon_{\mathrm{A1}}$.

Proof. Let $(\mathcal{G}, F_1, zF_1, dzF_1, azF_1, adzF_1, szF_1, F_2, zF_2, aF_2, xF_2, (dz - ax)F_2, Z_1)$ be the instance of A1 provided to $\mathscr{B}$. Let $Z_1 = c \cdot sdzF_1$. $\mathscr{B}$ has to determine whether $c = 1$ or $c \in_{\mathrm{U}} \mathbb{Z}_p$. The game is simulated as follows.

Set-Up: Pick $\alpha, v, v', y_1, \ldots, y_h, u \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and set the parameters as

$$P_1 = zF_1,\ V_2 = vF_2,\ V_2' = v'F_2,\ Q_{1,j} = y_j(dzF_1),\ U_1 = u(dzF_1),$$

$$aP_1 = azP_1,\ aQ_{1,j} = y_j(adzF_1),\ aU_1 = u(adzF_1),$$

where $j = 1, \ldots, h$ and similarly the elements τP_1, $\tau Q_{1,j}$ and τU_1. Compute $e(P_1, P_2)^\alpha = e(zF_1, zF_2)^\alpha$. $\mathscr{B}$ returns $\mathcal{PP}$ to $\mathscr{A}$. $\mathscr{B}$ knows $P_2 = zF_2$ and α but not $Q_{2,j}$'s and U_2. The main idea is to mask the components required to create identity-hash in $\mathbb{G}_2$ by a scalar multiple of aF_2 so that only semi-functional keys can be created.

Key Extraction Phases 1 and 2: $\mathscr{B}$ picks $w_1, w_2, r_1, r_2, r_3, r_4, (z_{1,j}, z_{2,j})_{j=1}^h \xleftarrow{\mathrm{U}} \mathbb{Z}_p$, $\gamma_1, \gamma_2 \xleftarrow{\mathrm{U}} \mathbb{Z}_p^\times$ and $\pi', (\pi_j')_{j=1}^h, \eta', (\eta_j')_{j=1}^h \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. It then computes the key for the i-th identity vector $\mathbf{id}_i = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell)$ as follows.

$$\begin{aligned}
&K_{1,1} = w_1(zF_2) + r_1V_2 - \gamma_1 aF_2,\ K_{1,2} = r_1V_2' + \gamma_1F_2,\ K_{1,3} = r_1F_2\\
&K_{2,1} = \alpha zF_2 + w_1h(\mathbf{id}_i)(dz - ax)F_2 + r_2V_2 - \gamma_1\pi'(aF_2),\ K_{2,2} = r_2V_2' + w_1h(\mathbf{id}_i)xF_2 + \gamma_1\pi'F_2,\ K_{2,3} = r_2F_2,
\end{aligned}$$

$$\begin{aligned}
&J_{1,1} = w_2(zF_2) + r_3V_2 - \gamma_2 aF_2,\ J_{1,2} = r_3V_2' + \gamma_2F_2,\ J_{1,3} = r_3F_2\\
&J_{2,1} = w_2h(\mathbf{id}_i)(dz - ax)F_2 + r_4V_2 - \gamma_2\eta'(aF_2),\ J_{2,2} = r_4V_2' + w_2h(\mathbf{id}_i)xF_2 + \gamma_2\eta'F_2,\ J_{2,3} = r_4F_2,
\end{aligned}$$

For $\ell + 1 \leq j \leq h$,

$$\begin{aligned}
&D_{j,1} = w_1y_j(dz - ax)F_2 + z_{1,j}V_2 - \gamma_1\pi_j'(aF_2),\ D_{j,2} = z_{1,j}V_2' + w_1y_j(xF_2) + \gamma_1\pi_j'F_2,\ D_{j,3} = z_{1,j}F_2\\
&E_{j,1} = w_2y_j(dz - ax)F_2 + z_{2,j}V_2 - \gamma_2\eta_j'(aF_2),\ E_{j,2} = z_{2,j}V_2' + w_2y_j(xF_2) + \gamma_2\eta_j'F_2,\ E_{j,3} = z_{2,j}F_2
\end{aligned}$$

setting $\pi = \pi' + \gamma_1^{-1}w_1h(\mathbf{id}_i)x$, $\pi_j = \pi_j' + \gamma_1^{-1}w_1y_jx$, $\eta = \eta' + \gamma_2^{-1}w_2h(\mathbf{id}_i)x$ and $\eta_j = \eta_j' + \gamma_2^{-1}w_2y_jx$. Since all these scalars are additively randomised they remain properly distributed in the adversary's view. We show that $D_{j,1}, D_{j,2}$ are well-formed; the rest can be verified in a similar fashion.

$$\begin{aligned}
D_{j,1} &= w_1 y_j(dz - ax)F_2 + z_{1,j}V_2 - \gamma_1\pi'_j(aF_2) \\
&= w_1 y_j dzF_2 - w_1 y_j axF_2 + z_{1,j}V_2 - \gamma_1(\pi_j - \gamma_1^{-1} w_1 y_j x)(aF_2) \\
&= w_1 y_j dzF_2 - w_1 y_j axF_2 + z_{1,j}V_2 - a\gamma_1\pi_j F_2 + w_1 y_j axF_2 \\
&= w_1 y_j dzF_2 + z_{1,j}V_2 - a\gamma_1\pi_j F_2 \\
D_{j,2} &= z_{1,j}V'_2 + w_1 y_j(xF_2) + \gamma_1\pi'_j F_2 \\
&= z_{1,j}V'_2 + w_1 y_j(xF_2) + \gamma_1(\pi_j - \gamma_1^{-1} w_1 y_j x)F_2 \\
&= z_{1,j}V'_2 + w_1 y_j xF_2 + \gamma_1\pi_j F_2 - w_1 y_j xF_2 \\
&= z_{1,j}V'_2 + \gamma_1\pi_j F_2
\end{aligned}$$

Challenge: $\mathscr{B}$ receives two pairs of messages and identity vectors $(M_0, \widehat{\mathbf{id}}_0)$ and $(M_1, \widehat{\mathbf{id}}_1)$ from $\mathscr{A}$. It chooses $\beta \xleftarrow{\mathrm{U}} \{0,1\}$ and $a', \xi \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ at random and generates a semi-functional challenge ciphertext as follows.

$C_0 \xleftarrow{\mathrm{U}} \mathbb{G}_T$
$C_{1,1} = h(\widehat{\mathbf{id}}_\beta)Z_1,\ C_{1,2} = a'h(\widehat{\mathbf{id}}_\beta)Z_1 + \xi F_1,\ C_{1,3} = -vh(\widehat{\mathbf{id}}_\beta)Z_1 - v'a'h(\widehat{\mathbf{id}}_\beta)Z_1 - v'\xi F_1,$
$C_{2,1} = szF_1,\ C_{2,2} = a'szF_1,\ C_{2,3} = -v(szF_1) - v'a'(szF_1),$

where $a' = a + \mu'$, $\mu = \mu' sz$ and $\xi = \mu\sigma'$. The challenge ciphertext $\widehat{\mathcal{C}} = (C_0, C_{1,1}, C_{1,2}, C_{1,3}, C_{2,1}, C_{2,2}, C_{2,3})$ is returned to $\mathscr{A}$. The computations below illustrate that $\widehat{\mathcal{C}}$ is a semi-functional encryption with $\sigma = \sigma' + cdh(\widehat{\mathbf{id}}_\beta)$.

$$\begin{aligned}
C_{1,2} &= a'h(\widehat{\mathbf{id}}_\beta)Z_1 + \xi F_1 \\
&= (a + \mu')h(\widehat{\mathbf{id}}_\beta)csdzF_1 + \mu\sigma' F_1 \\
&= ah(\widehat{\mathbf{id}}_\beta)csdzF_1 + \mu' h(\widehat{\mathbf{id}}_\beta)csdzF_1 + \mu\sigma' F_1 \\
&= as\mathcal{H}_1(\widehat{\mathbf{id}}_\beta) + (\mu' sz)(cdh(\widehat{\mathbf{id}}_\beta))F_1 + \mu\sigma' F_1 \\
&= as\mathcal{H}_1(\widehat{\mathbf{id}}_\beta) + \mu(cdh(\widehat{\mathbf{id}}_\beta))F_1 + \mu\sigma' F_1 \\
&= as\mathcal{H}_1(\widehat{\mathbf{id}}_\beta) + \mu\sigma F_1
\end{aligned}$$

Observe that $C_{1,1} = s\mathcal{H}_1(\widehat{\mathbf{id}}_\beta) = (c \cdot h(\widehat{\mathbf{id}}_\beta))(sdzF_1)$. If $c = 1$, then $\sigma = \sigma' + dh(\widehat{\mathbf{id}}_\beta)$ and $\widehat{\mathcal{C}}$ is encrypted under $\widehat{\mathbf{id}}_\beta$. Otherwise, c is random, causing $h(\widehat{\mathbf{id}}_\beta)$ and consequently the target identity and σ to be random quantities.
Guess: $\mathscr{A}$ returns its guess β' of β.
If the algorithm $\mathscr{B}$ returns 1 when $\beta = \beta'$ and 0 otherwise, it can solve the A1 instance with advantage

$$\mathsf{Adv}_{\mathcal{G}}^{\mathrm{A1}}(\mathscr{B}) = |\Pr[\beta = \beta' | Z_1 \text{ is real}] - \Pr[\beta = \beta' | Z_1 \text{ is random}]| = |\Pr[X_{M\text{-}rand}] - \Pr[X_{final}]|.$$

□

6 Conclusion

We have extended the Lewko-Waters IBE scheme using asymmetric pairings to a constant-size ciphertext HIBE. In addition to CPA-security the HIBE scheme possesses anonymity. Security is based on the assumptions LW1, LW2, DBDH-3 and a new assumption A1 that we introduce. This HIBE is the first example of an anonymous, adaptive-id secure, constant-size ciphertext HIBE which can be instantiated using Type-3 pairings. The assumptions used are static but non-standard. It would be interesting to explore constructions that obtain security under standard assumptions.

Note

A recent work by Lee, Park and Lee [13] proposes a construction identical to ours. Their proof of anonymity, however, relies on different assumptions namely – SXDH and asymmetric 3-party Diffie-Hellman (while our proof is based on A1). We would like to mention that this appeared in DCC August 2013 issue and was made publicly available after we submitted to IMACC 2013.

Acknowledgement

We thank the reviewers of IMACC 2013 for providing useful comments.

References

1. Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In Victor Shoup, editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 205–222. Springer, 2005.
2. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity-based encryption with constant size ciphertext. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer, 2005. Full version available at Cryptology ePrint Archive; Report 2005/015.
3. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003. Earlier version appeared in the proceedings of CRYPTO 2001.
4. Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307. Springer, 2006.
5. Sanjit Chatterjee and Alfred Menezes. On cryptographic protocols employing asymmetric pairings – the role of ψ revisited. *Discrete Applied Mathematics*, 159(13):1311–1322, 2011.
6. Clifford Cocks. An identity-based encryption scheme based on quadratic residues. In Bahram Honary, editor, *IMA Int. Conf.*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–363. Springer, 2001.
7. Angelo De Caro, Vincenzo Iovino, and Giuseppe Persiano. Fully secure anonymous hibe and secret-key anonymous ibe with short ciphertexts. In Marc Joye, Atsuko Miyaji, and Akira Otsuka, editors, *Pairing-Based Cryptography - Pairing 2010*, volume 6487 of *Lecture Notes in Computer Science*, pages 347–366. Springer Berlin / Heidelberg, 2010.
8. Léo Ducas. Anonymity from asymmetry: New constructions for anonymous hibe. In Josef Pieprzyk, editor, *CT-RSA*, volume 5985 of *Lecture Notes in Computer Science*, pages 148–164. Springer, 2010.
9. David Mandell Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 44–61. Springer, 2010.
10. Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.
11. Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography. In Yuliang Zheng, editor, *ASIACRYPT*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer, 2002.
12. Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 466–481. Springer, 2002.
13. Kwangsu Lee, JongHwan Park, and DongHoon Lee. Anonymous hibe with short ciphertexts: full security in prime order groups. *Designs, Codes and Cryptography*, pages 1–31, 2013.

14. Allison B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 318–335. Springer, 2012.
15. Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In Daniele Micciancio, editor, *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 455–479. Springer, 2010.
16. Jong Hwan Park and Dong Hoon Lee. Anonymous hibe: Compact construction over prime-order groups. *IEEE Transactions on Information Theory*, 59(4):2531–2541, 2013.
17. Jae Hong Seo, Tetsutaro Kobayashi, Miyako Ohkubo, and Koutarou Suzuki. Anonymous hierarchical identity-based encryption with constant size ciphertexts. In Stanislaw Jarecki and Gene Tsudik, editors, *Public Key Cryptography*, volume 5443 of *Lecture Notes in Computer Science*, pages 215–234. Springer, 2009.
18. Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.
19. Nigel P. Smart and Frederik Vercauteren. On computable isomorphisms in efficient asymmetric pairing-based systems. *Discrete Applied Mathematics*, 155(4):538–547, 2007.
20. Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, 2009.