Efficient (Anonymous) Compact HIBE From Standard Assumptions

Somindu C. Ramanna and Palash Sarkar

Applied Statistics Unit
Indian Statistical Institute
203, B. T. Road
Kolkata, India, PIN-700108
somindu_r@isical.ac.in palash@isical.ac.in

Abstract. We present two hierarchical identity-based encryption (HIBE) schemes, denoted as $\mathcal{H}_1$ and $\mathcal{H}_2$, from Type-3 pairings with constant sized ciphertexts. Scheme $\mathcal{H}_1$ achieves anonymity while $\mathcal{H}_2$ is non-anonymous. The constructions are obtained by extending the IBE scheme recently proposed by Jutla and Roy (Asiacrypt 2013). Security is based on the standard decisional Symmetric eXternal Diffie-Hellman (SXDH) assumption. In terms of provable security properties, previous direct constructions of constant-size ciphertext HIBE had one or more of the following drawbacks: security in the weaker model of selective-identity attacks; exponential security degradation in the depth of the HIBE; and use of non-standard assumptions. The security arguments for $\mathcal{H}_1$ and $\mathcal{H}_2$ avoid all of these drawbacks. These drawbacks can also be avoided by obtaining HIBE schemes by specialising schemes for hierarchical inner product encryption; the downside is that the resulting efficiencies are inferior to those of the schemes reported here. Currently, there is no known anonymous HIBE scheme having the security properties of $\mathcal{H}_1$ and comparable efficiency. An independent work by Chen and Wee describes a non-anonymous HIBE scheme with security claims and efficiency similar to that of $\mathcal{H}_2$; we note though that in comparison to $\mathcal{H}_2$, the Chen-Wee HIBE scheme has larger ciphertexts and less efficient encryption and decryption algorithms. Based on the current state-of-the-art, $\mathcal{H}_1$ and $\mathcal{H}_2$ are the schemes of choice for efficient implementation of (anonymous) HIBE constructions.

Keywords: hierarchical identity-based encryption (HIBE); constant-size ciphertext HIBE; asymmetric pairings; standard assumptions; dual-system encryption.

1 Introduction

Identity-based encryption (IBE) is a form of public key encryption where a recipient's identity itself is her public key. The corresponding decryption key is generated and securely transmitted by a trusted authority called private key generator (PKG). The concept of IBE was introduced by Shamir [42] and the first constructions were proposed in [18, 5]. In order to reduce the communication and computation overhead on the PKG, the notion of hierarchical IBE ([25, 26]) was introduced. HIBE imposes a tree-like structure on entities within the system and provides the higher level entities the ability to delegate key generation to lower-level entities without the involvement of the PKG.

This work presents two new HIBE schemes called $\mathcal{H}_1$ and $\mathcal{H}_2$. The literature already contains several different HIBE schemes. So, the question arises as to why new ones are needed? We argue below that previous direct constructions of HIBE schemes had one or more drawbacks related to either efficiency or security. The new schemes overcome all these issues and are the candidates of choice for any practical deployment. To understand this, we need to discuss the different efficiency and security issues that arise while constructing HIBE systems.

Efficiency. Practical constructions of HIBE schemes are obtained from *pairings*. A pairing is a bilinear, non-degenerate and efficiently computable map e from $\mathbb{G}_1 \times \mathbb{G}_2$ to $\mathbb{G}_T$, where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups of the same order. Practical instantiations of such maps are obtained 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.

Type-3 Pairings: There are different types of pairings which can be obtained from elliptic curves. Pairings where the common group order is prime and it is computationally infeasible to find an isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$ are called Type-3 pairings. Such pairings have the most efficient implementations, both in terms of computation and representation [9, 44, 23]. 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).

Constant-Size Ciphertexts: Recall that in a HIBE scheme, an individual entity can obtain a private key from either the PKG or from a lower-level entity. In the later case, the complete identity of the entity is obtained by appending its individual identity to the identity of the entity from which it obtains the private key. As a result, identities in a HIBE set-up are variable length tuples of strings. For some HIBE schemes, including the initial ones [25] and later works [3, 45, 10, 46, 39], the length of a ciphertext grows linearly with the length of the identity. As a result, an encryption to an entity which is further away from the PKG incurs a communication penalty compared to an encryption to an entity which is closer to the PKG. In practical terms, such artificial asymmetry in communication overhead is undesirable. The solution to this is to have a HIBE scheme where the size of the ciphertext is independent of the length of the identity tuple. We refer to such schemes as constant-size ciphertext HIBE (denoted CC-HIBE) schemes.

The discussion above suggests that from an efficiency point of view, HIBE schemes with *constant-size ciphertexts* that can be instantiated with *Type-3 pairings* would offer the best performances.

The first construction for CC-HIBE was given by Boneh, Boyen and Goh [4]. This work introduced a way to hash identity vectors into the pairing groups. Almost all known CC-HIBE schemes that appeared later have either used this technique or a variant [11, 12, 32, 41, 20, 37, 29, 40, 14]. Since we are interested in CC-HIBE, we do not consider the line of work [25, 3, 45, 10, 46, 39] where the length of the ciphertext depends on the length of the identity tuple. Another method for obtaining constant-size ciphertext HIBE is to specialise constructions of hierarchical inner product encryption. The resulting schemes are not efficient. We comment on this later.

Security. Several security-related issues crop up while building HIBE schemes.

Security Model: In terms of security, the goal is to obtain (H)IBE schemes which are secure against adaptive-identity attacks [5, 25, 43]. The first HIBE construction [25] (though not a CC-HIBE) indeed achieved this, but, the security argument was based on the use of random oracles. Later works could avoid the use of random oracles, but, some of them could only be proved secure in the much weaker model of selective-identity attacks. The first CC-HIBE scheme [4] is secure only under this weaker model.

Anonymity: Another important security notion is *anonymity* [1] which requires that a ciphertext does not reveal any information about the recipient's identity. Anonymous (H)IBE schemes are useful in constructing public key encryption with keyword search (PEKS) which further extends to more sophisticated primitives such as public key encryption with temporary keyword search (PETKS) and identity-based encryption with keyword search (IBEKS) [1].

Hardness Assumptions: Security proofs are essentially reductionist arguments that are based on the assumption that some problems are computationally hard to solve. Certain problems, such as the decisional Diffie-Hellman (DDH) problem over appropriate groups have widespread use in cryptography. Other examples are the decisional bilinear Diffie-Hellman (DBDH) problem, the decisional linear (DLin) problem and the decisional symmetric external Diffie-Hellman (SXDH) problem. Schemes whose security is based on the hardness of such problems are said to be based on standard assumptions. In contrast, certain schemes are based on less well-studied problems which are tailor-made to suit the requirements

of the particular scheme. These assumptions are referred to as non-standard. Further, these assumptions are sometimes parametrised by a quantity arising in the construction (e.g. maximum length of an identity tuple, number of key extract queries). Such assumptions are called non-static.

Degradation: Proofs of security (for HIBE schemes) are reductions of the following form. If an algorithm running in time t breaks the security of the scheme with "advantage" ε, then some computational problem Π can be solved in time t' with advantage ε'. The ratio δ of t'/ε' to t/ε is the tightness gap and the reduction is said to have a degradation of δ. Depending upon the scheme and the reduction, δ could be a constant or could depend on quantities such as the security parameter, the maximum number of corrupt users, the maximum length of an identity tuple and possibly other parameters. Designing schemes that have low degradation is important.

Prior to [24] all HIBE schemes suffered from a degradation which is exponential in the depth of the HIBE. The construction in [24] is very complicated and the security is based on an unnatural assumption. The first practical method of constructing HIBE schemes where security does not degrade with the depth of the HIBE is due to Waters [46] who introduced the very important technique of dual-system encryption. The work [32] provided the first CC-HIBE scheme based on composite-order pairings following the dual-system approach.

Prefix Decryption. In some HIBE schemes, a ciphertext for an identity vector can be decrypted by any entity possessing a secret key for a prefix of that identity. Let us name this property *prefix decryption*. In constructions with separate ciphertext elements corresponding to individual components of the identity tuple, such as the one in [25], prefix decryption is facilitated – the ciphertext can be truncated to obtain a valid ciphertext under the prefix identity vector and thus can be decrypted using the corresponding key. Prefix decryption in a CC-HIBE could be done as follows: the key for the prefix is used to create a key corresponding to the recipient identity via delegation; the resulting key is used to decrypt the ciphertext. The latter method fails when the scheme is anonymous i.e., when the recipient identity is hidden. Delegation can no longer be done without knowledge of the recipient identity. The above discussion suggests that achieving constant-size ciphertexts and anonymity simultaneously results in the loss of prefix decryption. Although it may seem that this restriction is somehow tied to the property of anonymity, we would like to emphasise that this is a definitional issue. That is, whether prefix decryption is allowed or not must reflect in the HIBE definition. The definition we provide does not explicitly allow prefix decryption. We stress that the prefix decryption property is absent in all known HIBE constructions that concurrently attain constant-size ciphertexts and anonymity. Furthermore, all known HIBE schemes possess at most two out of the three features – constant-size ciphertexts, anonymity and prefix decryption.

On the other hand, not having prefix decryption guards against key escrow to some extent. An entity has the power to delegate keys to lower level entities but cannot decrypt ciphertexts sent to the lower-level entities. This feature may be useful in applications such as email. If the higher level entities are key generating servers, user privacy will not be compromised in the event that any of these servers is corrupted. Further, in primitives such as identity-based searchable encryption obtained from anonymous HIBE schemes [1], this limitation does not make any difference.

We provide a more detailed discussion on prefix decryption in Section 4.2.

1.1 Possible Approaches to the Construction of HIBE Schemes.

We have argued above that among HIBE schemes, it is CC-HIBE which is of practical importance and among the known CC-HIBE schemes, $\mathcal{H}_1$ and $\mathcal{H}_2$ are the most suitable ones for practical deployment. As mentioned earlier, both schemes are based on the recently proposed IBE due to Jutla and Roy [28] (abbreviated JR-IBE).

Extension from IBE. It is quite natural that the construction of a HIBE scheme will be based on an IBE scheme. Below we list other candidate IBE schemes and why their extensions to HIBE schemes do not achieve the same security and efficiency as $\mathcal{H}_1$ or $\mathcal{H}_2$.

To start with, it is desirable to avoid a security degradation which is exponential in the depth of the HIBE. In the current state of the art, this means that one has to follow the dual-system approach. So, any attempt to construct a CC-HIBE should start with an IBE which has been proved secure using the dual-system technique. In the dual-system proof technique for both IBE and HIBE, ciphertext and key in the scheme itself are called *normal*. As part of the proof, alternate forms of ciphertext and key are defined. These are called *semi-functional*. In the proof, these are simulated using instances of some hard problem and the argument proceeds by showing that an adversary's ability to distinguish between normal and semi-functional components can be translated into an algorithm to solve the problem. During simulation, it is essential to ensure that all ciphertexts and keys given to the attacker including the semi-functional components (possibly generated using elements from the problem instance) have proper distributions i.e., as in the real construction. This requirement creates the main hurdle in extending the known IBE schemes with dual system proofs to CC-HIBE while retaining the security properties. We dicuss this problem below in detail.

The IBE constructions of Waters [46] and its variants [39] do not have a structure that is suitable for extension to CC-HIBE. This is because both the ciphertext and keys have associated *tags* that are public and play a crucial role in dual system arguments. It is precisely these tags that cause the problem in extending these IBEs to CC-HIBEs. While extending to a CC-HIBE, sufficient information should be provided in either the public parameters or the keys to support rerandomisation during key delegation. The tags either cannot be rerandomised or the elements needed to enable their rerandomisation, when given out, lead to insecure schemes.

Lewko and Waters [32] presented a new variant of dual system technique by shifting the role of tags into the semi-functional components. This enabled them to obtain a CC-HIBE scheme over composite order pairing groups. They converted the IBE version of the scheme to the prime-order asymmetric pairing setting but not the HIBE scheme. Security of both their IBE schemes (for composite-order pairings as well as for Type-3 pairings) are based on static but non-standard assumptions. Two works [29, 40] independently obtained a CC-HIBE scheme from Lewko-Waters' IBE in prime-order groups. Both the schemes are anonymous and achieve security under static assumptions. The only drawback is that the assumptions are non-standard.

Another IBE scheme following the dual-system approach is due to Chen *et.al.* [13]. This work uses *dual pairing vector spaces* (DPVSs) [33, 34]. These are algebraic structures that have properties found in composite order groups such as *cancelling* and *parameter-hiding* which are useful for dual system arguments [31]. The Chen *et.al.* IBE can be seen as a translation of Lewko-Waters' composite-order pairing-based IBE [32] to the setting of asymmetric pairing using DPVS. It is then natural to ask whether the Lewko-Waters composite-order CC-HIBE can be similarly translated using the DPVS-approach to a CC-HIBE. Unfortunately, such a transformation does not yield a CC-HIBE. This is due to the fact that for the proof to work, the dimension of the vector spaces becomes proportional to the HIBE depth. Since ciphertexts contain vectors from such spaces, the constant-size feature cannot be attained.

Chen and Wee [15] introduced new techniques for parameter-hiding in DPVS-based constructions. The paper describes an IBE scheme and provides a detailed security proof. Both the conference version and a later eprint version (with the same title) of the paper mention the construction of a compact HIBE scheme. None of these versions, however, provide the actual construction of the HIBE scheme or any other details. Another paper [16] by the same authors provide the construction of the HIBE scheme and the security proof. We note though that our work has been done independently and without seeing or knowing

about the construction described in [16][1]. Apart from the issue of the independence of our work with that in [16], there are differences between the two works. First, the approach of the present work is different from that of [16]; second, we provide both an anonymous HIBE scheme ($\mathcal{H}_1$) and a non-anonymous HIBE scheme ($\mathcal{H}_2$), whereas [16] provides only a non-anonymous HIBE scheme; and third, compared to the non-anonymous HIBE scheme in [16], $\mathcal{H}_2$ has smaller ciphertexts and more efficient encryption and decryption algorithms.

Hierarchical Inner-Production Encryption. HIBE schemes can also be seen as special cases of hierarchical inner product encryption (HIPE) schemes. HIBE schemes obtained from two constructions of HIPE schemes by Okamoto and Takashima [35, 36] are comparable to our schemes in terms of provable properties. The scheme in [35] is not anonymous but achieves constant-size ciphertexts. The construction in [36] achieves anonymity and prefix decryption but not constant-size ciphertexts. Note that none of the two HIBE schemes achieve all three properties (anonymity, constant-size ciphertexts and prefix decryption) at the same time. Both schemes are based on dual pairing vector spaces over symmetric pairing groups and are shown to be secure under decisional linear (DLin) assumptions. Although constructions based on DPVSs can be extended to more sophisticated primitives such as attribute-based encryption, a drawback of using this approach is that the ciphertext size depends on the dimension of some DPVS (chosen during system setup). This restricts us from making any further optimisations on the size of the ciphertexts by "transfering" some structure to the keys. Another drawback is as follows. Since the HIBE is an instantiation of (zero) inner-product encryption, the length of the (attribute-)vector consisting of the identity has to be twice the maximum depth (h) of the hierarchy to enable cancellation. Moreover, to accommodate a dual system proof, the dimension of the vector space needs to be at least twice this length i.e., $4h$. This will result in larger keys. Needless to say, we may hope to obtain more efficient HIBE schemes through direct constructions.

Predicate Encryption. Any predicate encryption (PE) scheme is defined by a predicate $\mathcal{P}$. A ciphertext in a PE scheme is associated to an attribute x in addition to the message that it encrypts. Decryption by a secret key associated to attribute y succeeds if and only if $\mathcal{P}(x, y) = 1$. Many primitives such as IBE, spatial encryption and inner product encryption can be viewed as specific cases of predicate encryption. In particular, a HIBE scheme is a PE scheme defined by the equality predicate over hierarchical identities. Here, the attributes associated with the ciphertexts and keys are nothing but the hierarchical identities.

Recent works by Wee [47] and Attrapadung [2] provide generic constructions of predicate encryption schemes achieving full security via dual system techniques. Both the works use composite order groups for their construction. As mentioned earlier, in principle it should be possible to obtain HIBE schemes in Type-3 setting from these works. Doing this would require specialising a PE scheme to a HIBE scheme and also converting from composite-order setting to the prime-order setting possibly by using the tools from [31, 13]. It is not clear though that the resulting HIBE schemes will have efficiencies comparable to that of $\mathcal{H}_1$ or $\mathcal{H}_2$. We believe that HIBE is an important enough primitive to warrant research on obtaining direct and efficient constructions of such schemes.

1.2 Extending JR-IBE to CC-HIBE

Schemes $\mathcal{H}_1$ and $\mathcal{H}_2$ extend the JR-IBE to anonymous and non-anonymous CC-HIBEs respectively. At a top level, the identity-hashing technique of Boneh-Boyen-Goh [4] (BBG-hash) is applied on JR-IBE. We work in the setting of asymmetric pairings where ciphertext components are elements of $\mathbb{G}_1$ and key

[1] We can provide a chronology of events that justify that our work has in fact been done independently of [16].

components are elements of $\mathbb{G}_2$. BBG-hash of the identity is required to be computed in both $\mathbb{G}_1$ and $\mathbb{G}_2$. During encryption, the BBG-hash is required to be computed in $\mathbb{G}_1$ and this requires adding some elements of $\mathbb{G}_1$ to the public parameters.

In previous CC-HIBE schemes in the prime-order setting within the dual system framework [29, 40], anonymity appears as a by-product of the HIBE extension. The basic difficulty in making it non-anonymous was due to the following dichotomy concerning key delegation. The BBG-hash for the key is computed in $\mathbb{G}_2$. The hash is defined using certain elements of $\mathbb{G}_2$. During key delegation, the hash has to be rerandomised and so the elements should be publicly available. On the other hand, information about these elements must not be leaked because they form the source of randomness used to generate the semi-functional components during simulation.

The problem described above does not arise in case of JR-IBE. The feature of JR-IBE that makes extension to the non-anonymous CC-HIBE $\mathcal{H}_2$ possible is as follows. The master secret consists of two elements whose linear combination is used to mask the message during encryption. This is unlike previous (H)IBE schemes where a single element was used for the purpose. The two elements would be information theoretically hidden from an attacker's view. So the secret randomness for the semi-functional ciphertext space is provided by one of the two elements.

Anonymity is achieved by keeping the elements required to compute the BBG-hash in $\mathbb{G}_2$ to be secret and instead provide suitably randomised copies of these elements in the user keys. Problems then arise while defining *semi-functional* components and arguing about their well-formedness during simulation. Fortunately, it turns out that the problems can be handled by using appropriate algebraic relations. The technique of keeping certain elements hidden and providing their randomised version in the user keys closely follow the ideas introduced in [6] to obtain anonymity. In $\mathcal{H}_1$ the elements that are kept hidden are exactly the ones required to create the BBG-hash in $\mathbb{G}_2$. As a result, an adversary is unable to create an identity hash in $\mathbb{G}_2$ and cancel it out with the BBG-hash of the same identity in $\mathbb{G}_1$. This naturally leads to the scheme $\mathcal{H}_1$ being anonymous.

We note that a single-level instantiation of $\mathcal{H}_2$ provides a non-anonymous variant of the JR-IBE with rerandomisable keys.

1.3 Detailed Comparison to Existing HIBE Schemes.

Table 1 provides a comparison of $\mathcal{H}_2$ with all previously proposed non-anonymous CC-HIBE schemes. In terms of security, $\mathcal{H}_2$ is comparable to [35] and [14]. The security of the construction in [32] is based on sub-group decision assumptions that cannot be considered to be standard assumptions. $\mathcal{H}_2$ achieves the best efficiency compared to all other schemes. Table 2 compares $\mathcal{H}_1$ with all previously proposed anonymous HIBE schemes. In terms of security and efficiency, there is no construction that is comparable to $\mathcal{H}_1$.

We fix some notation required to compare different parameters of HIBE constructions. h: maximum depth of the HIBE; ℓ: length of the identity tuple; q: number of key extraction queries. In [11], N is the number of bits in an identity and k represents number of blocks of N/k bits. #pp, #msk, #cpr and #key denote number of group elements in the public parameters, master secret, ciphertext and key respectively. Enc, Dec, KGen and Deleg indicate the efficiency of encryption, decryption, key generation and delegation algorithms. For Type-3 pairing based schemes, $\mathcal{PP}$ and ciphertexts consist elements of $\mathbb{G}_1$; $\mathcal{MSK}$ and keys consist elements of $\mathbb{G}_2$. #pp $= (a, b)$ means that there are a elements from $\mathbb{G}_1$, $\mathbb{G}_2$ and b elements of $\mathbb{G}_T$. #cpr $= (a, b)$ denotes a elements from $\mathbb{G}_1$ and b elements from $\mathbb{Z}_p$ where $p = |\mathbb{G}_1|$. We do not consider the $\mathbb{G}_T$ element that masks the message in our comparison as it is present in all constructions. Enc $= (a, b)$ implies that a scalar multiplications are required in $\mathbb{G}_1$ and b exponentiations in $\mathbb{G}_T$; 'Dec' is measured in terms of number of pairings; 'KGen' is determined by number of scalar multiplications in $\mathbb{G}_2$; 'Deleg', by

number of scalar multiplications in $\mathbb{G}_2$. 'Assump' denotes the set of underlying complexity assumptions; Deg is a shorthand for security degradation. 'Prefix Dec' indicates whether or not the HIBE supports prefix decryption. 'Const #cpr' denotes constant number of elements in the ciphertext (or constant-size ciphertext).

Scheme	[4]	[11]	[12]	[32]	[35]	[14]	$\mathcal{H}_2$
Pairing	Type-1	Type-1	Type-1	Composite	Type-1	Type-3	Type-3
Security	selective-id	adaptive-id	selective$^+$-id	adaptive-id	adaptive-id	adaptive-id	adaptive-id
Assump.	Decisional h-wBDHI	h-wDBDHI*	h-wDBDHI*	Subgroup Decision	DLin	d-Lin	SXDH
Deg.	1	$O((kq2^{N/k})^h)$	1	$O(q)$	$O(q)$	$O(q)$	$O(q)$
#pp	$(h+4, 0)$	$(h+3+hk, 0)$	$(2h+3, 1)$	$(h+3, 1)$	$(32h^2+16h+25, 1)$	$(2d(d+1)(h+2), d)$	$(3h+9, 1)$
#msk	1	1	1	1	5	$d+1$	2
#cpr	(2,0)	(2,0)	(3,0)	(2,0)	(13,0)	(2(d+1),0)	(3,1)
#key	$h-\ell+2$	$(k+1)(h-\ell)+2$	$2(h-\ell+1)$	$h-\ell+2$	$8h+5$	$(d+1)(h-\ell+2)$	$2(h-\ell)+5$
Enc	$(\ell+2, 1)$	(2,1)	$(\ell+2, 1)$	$(\ell+2, 1)$	$32h+23$	$(d(d+1)(\ell+2), d)$	$(\ell+4, 1)$
Dec	2	2	2	2	13	$2(d+1)$	3
KGen	$h+2$	$2(h-\ell+1)$	$2h-\ell+2$	$2h-\ell+4$	$16h(h+\ell)+10$	$d(d+1)(h+2)$	$2h+7$
Deleg.	$\ell+2$	$2(h-\ell)$	$2h-\ell+1$	$2h-\ell+6$	$16h(h+\ell+1)+10$	$d(d+1)(h+2)+d+1$	$2h+9$

Table 1: Comparison of non-anonymous CC-HIBE schemes based on pairings without random oracles.

Efficiency comparison of $\mathcal{H}_2$ with [32]. In absolute terms, the number of group elements required for composite-order based schemes is less than that required in the new HIBE schemes. However, only counting group elements is not a proper comparison. One has to consider the actual size for representing a single group element at a desired security level.

For concreteness, let us consider a security level of 128 bits. For Type-3 pairings, using Table-2 of [8], elements of $\mathbb{G}_1$ and $\mathbb{G}_2$ can be represented using 257 and 513 bits respectively. In contrast, the order of $\mathbb{G}_1 = \mathbb{G}_2$ for composite-order pairings is a product of at least three primes. The basic security requirement is that this group order should be hard to factor. To attain 128-bit security level, the length of the bit representation of the group order should be about 3000 bits (or more). So, for schemes based on composite-order groups, the length of representations of elements of $\mathbb{G}_1$ (and $\mathbb{G}_2$) will be about 3000 bits. This is about 12 times (resp. 6 times) more than the length of bit representation of elements of $\mathbb{G}_1$ (resp. $\mathbb{G}_2$) using Type-3 pairings. The wide difference in the length of representations of group elements more than adequately compensates for the absolute number of group elements in composite-order HIBE schemes being lesser than that in the newly proposed HIBE scheme.

For example, ciphertexts in $\mathcal{H}_1$ (or $\mathcal{H}_2$) consist of 3 elements of $\mathbb{G}_1$ which is about 770 bits whereas ciphertexts in the HIBE of [32] will be about 9000 bits (3 elements each having length about 3000 bits). Similar considerations apply to public parameters ($\mathcal{PP}$), master secret key ($\mathcal{MSK}$) and decryption keys. The larger length of the parameters also lead to a significant slow down in the basic operations of scalar multiplication and pairing computation leading to much slower algorithms for encryption, decryption, key generation and key delegation.

Comparing $\mathcal{H}_2$ with [35] and [14]. The schemes in [35, 14] both achieve similar security guarantees as $\mathcal{H}_2$. The construction of [35] the number of elements in the ciphertext and the number of pairings required for decryption is 13 as opposed to just 3 in $\mathcal{H}_2$. The scheme in [14] achieves similar parameters when $d = 1$ (d-Lin is XDH when $d = 1$) but is still less efficient compared to $\mathcal{H}_2$ in terms of ciphertext size and

decryption time. Ciphertext in [14] will consist of 4 $\mathbb{G}_1$-elements whereas $\mathcal{H}_2$ contains 3 $\mathbb{G}_1$-elements along with an element of $\mathbb{Z}_p$. If an element of $\mathbb{G}_1$ is represented using two elements of $\mathbb{Z}_p$, then $\mathcal{H}_2$ ciphertexts consist of 7 $\mathbb{Z}_p$ elements as opposed to 8 in [14]. Certainly, $\mathcal{H}_2$ has shorter ciphertexts.

From Table 1 and the previous discussion, the only non-anonymous HIBE scheme which is comparable in efficiency and security to $\mathcal{H}_2$ is the Chen-Wee scheme described in [14] for $d = 1$ whence d-Lin becomes DDH. $\mathcal{H}_2$ has shorter ciphertexts and faster encryption and decryption algorithms, while the Chen-Wee scheme has shorter decryption keys and faster key generation and delegation algorithms. For an encryption scheme, encryption and decryption will be used more often than key generation and delegation, so, the advantage of $\mathcal{H}_2$ over the Chen-Wee scheme outweighs the disadvantages. When $d > 1$, the Chen-Wee scheme is based on progressively weaker assumptions than the SXDH assumption and the resulting schemes also become progressively more inefficient.

Scheme	[6]	[41]	[20]	[37]	[29],[40]	[36]	$\mathcal{H}_1$
Pairing	Type-3	Composite	Composite	Type-1	Type-3	Type-1	Type-3
Security	selective-id	selective-id	adaptive-id	selective-id	adaptive-id	adaptive-id	adaptive-id
Assump.	DLin,DBDH	ℓ-wBDH*, ℓ-cDH	Subgroup Decision	h-BDHE Aug. h-DLin	LW1,LW2,DBDH [29]:3-DH,XDH [40]:A1	DLin	SXDH
Deg.	$O(1)$	$O(1)$	$O(q)$	$O(1)$	$O(q)$	$O(hq)$	$O(q)$
Prefix Dec.	No	No	No	No	No	Yes	No
Const #cpr	No	Yes	Yes	Yes	Yes	No	Yes
#pp	$(2(h^2+3h+2),1)$	$(h+6,1)$	$(h+4,1)$	$(h+6,1)$	$(3h+6,1)$	$(4(9h+4),1)$	$(h+4,1)$
#msk	h^2+5h+7	$h+4$	2	4	$h+6$	$18h+10$	$2h+6$
#cpr	$(2h+5,0)$	(3,0)	(2,0)	(4,0)	(6,0)	$(9\ell+5,0)$	(3,1)
#key	$(h+3)(3h-\ell+5)$	$3(h-\ell+3)$	$2(h-\ell+2)$	$3(h-\ell+4)$	$6(h-\ell+2)$	$(4h-2\ell+1)(9\ell+5)+36(h-\ell)$	$4(h-\ell)+10$
Enc	$(2(\ell+3)(h+2)+1,1)$	$(\ell+6,1)$	$(\ell+4,1)$	$(\ell+5,1)$	$(3(\ell+2),1)$	$27\ell+15$	$(\ell+4,1)$
Dec	$2h+3$	4	2	4	6	$9\ell+5$	3
KGen	$h^3+h^2(5-\ell)+$ $h(7-3\ell)-2\ell+2$	$3h-2\ell+2$	$4(h+2-3\ell)$	$(h+2(h-\ell+8))$	$6h-5\ell+12$	$(2h+3)(27\ell+10)$	$2(2h-2\ell+5)$
Deleg.	$5(h+2)(h+3)+1$	$6(h-\ell)+21$	$4(h-\ell)+11$	$(4(h-\ell)+25)$	$2(h-\ell+3)$	$(9\ell+5)(6h\ell+14h-2\ell^2-8\ell+5)$	$4(h-\ell+5)$

Table 2: Comparison of anonymous HIBE schemes based on pairings without random oracles.

$\mathcal{H}_1$ and other anonymous HIBE schemes. It is clear from Table 2 that all anonymous HIBE schemes possess either constant-size ciphertexts or the prefix decryption property and not both. The Boyen-Waters HIBE [6] has none of the two properties. The Okamoto-Takashima scheme [36] supports prefix decryption and at the same time achieves anonymity but at the cost of non-constant size of the ciphertext (the size is linear in the depth of the identity). In addition, ciphertexts in their scheme reveal the length of the recipient identity unlike the Boyen-Waters HIBE. $\mathcal{H}_1$, on the other hand, is anonymous and has short ciphertexts but lacks prefix deryption. All other efficiency parameters are better in case of $\mathcal{H}_1$.

We conclude that among anonymous HIBE schemes, $\mathcal{H}_1$ is the most efficient scheme with all the standard provable properties. We emphasise that the efficiency and provable security properties achieved for $\mathcal{H}_1$ have not been simultaneously achieved earlier, either for composite-order pairings, or, for prime-order pairings. For use in practice, one may choose the Okamoto-Takashima scheme or $\mathcal{H}_1$ according to whether the application requires prefix decryption or not.

A note on notation and proof technique. We have used the JR-IBE [28] as the basic building block and consequently, our notation and proofs build on that of [28]. This makes it easier for a reader to see the connections between our work and the IBE construction in [28]. We note, though, that we have provided all the relevant details and it is possible to directly verify, with a bit of work, all the claims in

this paper without referring to [28]. Frameworks for presenting dual-system constructions and proofs have been proposed [31, 22]. Neither the JR-IBE nor the constructions in the present work appear to fall within these frameworks.

2 Preliminaries

Some basic notation, definitions and the complexity assumptions used in our proofs are presented in this section. Definition of HIBE and security notions are provided in Appendix A.

2.1 Notation

The notation $x_1, \ldots, x_k \xleftarrow{\mathrm{R}} \mathcal{X}$ indicates that elements $x_1, \ldots, x_k$ are sampled independently from the set $\mathcal{X}$ according to some distribution R. The uniform distribution is denoted U. For a (probabilistic) algorithm $\mathcal{A}$, $y \longleftarrow \mathcal{A}(x)$ means that y is chosen according to the output distribution of $\mathcal{A}$ on input x. $\mathcal{A}(x; r)$ denotes that $\mathcal{A}$ is run on input x with its internal random coins set to r. For two integers $a < b$, the notation $[a, b]$ represents the set $\{x \in \mathbb{Z} : a \leq x \leq b\}$. If $\mathbb{G}$ is a finite cyclic group, then $\mathbb{G}^{\times}$ denotes the set of generators of $\mathbb{G}$.

2.2 Asymmetric Pairings and Hardness Assumptions

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 groups written additively and $\mathbb{G}_T$ is a multiplicatively written group, all having the same order p and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a map with the following properties.

1. *Bilinearity:* 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-degeneracy:* 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. *Efficient computation:* The function e should be efficiently computable.

In an asymmetric pairing, $\mathbb{G}_1 \neq \mathbb{G}_2$. If no efficiently computable isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$ are known, then such pairings are called Type-3 pairings. The terms 'Type-3 pairing' and 'asymmetric pairing' are used interchangeably in the rest of the paper.

Let $\mathcal{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ be an asymmetric pairing and $\mathscr{A}$, a probabilistic polynomial time (PPT) algorithm $\mathscr{A}$ that outputs 0 or 1. We now describe the decisional Diffie-Hellman (DDH) assumptions in groups $\mathbb{G}_1$ and $\mathbb{G}_2$, called DDH1 and DDH2 respectively.

Assumption DDH1. Define a distribution $\mathcal{D}$ as follows: $P_1 \xleftarrow{\mathrm{U}} \mathbb{G}_1^{\times}$; $P_2 \xleftarrow{\mathrm{U}} \mathbb{G}_2^{\times}$, $a, s \xleftarrow{\mathrm{U}} \mathbb{Z}_p$, $\mu \xleftarrow{\mathrm{U}} \mathbb{Z}_p$: $\mathcal{D} = (\mathcal{G}, P_1, aP_1, asP_1)$. The advantage of $\mathscr{A}$ in solving the DDH1 problem is given by

$$\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}}(\mathscr{A}) = |\Pr[\mathscr{A}(\mathcal{D}, sP_1) = 1] - \Pr[\mathscr{A}(\mathcal{D}, (s+\mu)P_1) = 1]|.$$

Essentially, $\mathscr{A}$ has to decide whether $\mu = 0$ or $\mu \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ given $(\mathcal{D}, (s+\mu)P_1)$. The (ε, t)-DDH1 assumption holds in $\mathcal{G}$ if for any adversary $\mathscr{A}$ running in time at most t, $\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}}(\mathscr{A}) \leq \varepsilon$.

Assumption DDH2. Let a distribution $\mathcal{D}$ be defined as follows: $P_1 \xleftarrow{U} \mathbb{G}_1^\times$; $P_2 \xleftarrow{U} \mathbb{G}_2^\times$, $r, c \xleftarrow{U} \mathbb{Z}_p$, $\gamma \xleftarrow{U} \mathbb{Z}_p$: $\mathcal{D} = (\mathcal{G}, P_1, P_2, rP_2, cP_2)$. $\mathscr{A}$'s advantage in solving the DDH2 problem is given by

$$\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH2}}(\mathscr{A}) = |\Pr[\mathscr{A}(\mathcal{D}, rcP_2) = 1] - \Pr[\mathscr{A}(\mathcal{D}, (rc+\gamma)P_2) = 1]|.$$

The (ε, t)-DDH2 assumption is that, for any t-time algorithm $\mathscr{A}$, $\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH2}}(\mathscr{A}) \leq \varepsilon$.

3 Jutla-Roy IBE with Ciphertexts in $\mathbb{G}_1$

In the IBE scheme of Jutla-Roy [28] (JR-IBE), ciphertext consists of elements in $\mathbb{G}_2$ and keys contain elements from $\mathbb{G}_1$. For Type-3 pairings, elements of $\mathbb{G}_1$ have a shorter representation compared to the elements of $\mathbb{G}_2$. To reduce the length of the ciphertext, one has to interchange the roles of the two groups. In contrast, for a signature scheme, it would be advantageous to have the signature to consist of elements from $\mathbb{G}_1$. Since the JR-IBE is obtained from non-interactive zero knowledge (NIZK) proofs via the idea of signatures, the scheme results in ciphertext elements being in $\mathbb{G}_2$.

This section describes a "dual" of the Jutla-Roy [28] (JR-IBE-D) where ciphertexts live in $\mathbb{G}_1$ and keys in $\mathbb{G}_2$. 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.

Parameters: Choose $P_1 \xleftarrow{U} \mathbb{G}_1^\times$, $P_2 \xleftarrow{U} \mathbb{G}_2^\times$, $\Delta_1, \Delta_2, \Delta_3, \Delta_4, c, d, u, e \xleftarrow{U} \mathbb{Z}_p$, $b \xleftarrow{U} \mathbb{Z}_p^\times$, and set $U_1 = (-\Delta_1 b + d)P_1$, $V_1 = (-\Delta_2 b + e)P_1$, $W_1 = (-\Delta_3 b + c)P_1$, $g_T = e(P_1, P_2)^{-\Delta_4 b + u}$. The parameters are given by

$\mathcal{PP} : (P_1, bP_1, U_1, V_1, W_1, g_T)$
$\mathcal{MSK} : (P_2, cP_2, \Delta_1, \Delta_2, \Delta_3, \Delta_4, d, u, e)$

Ciphertext: Consists of $(C_0, C_1, C_2, C_3, \mathsf{tag})$ where

$\mathsf{tag}, s \xleftarrow{U} \mathbb{Z}_p$, $\{\mu \xleftarrow{U} \mathbb{Z}_p\}$
$C_0 = m \cdot (g_T)^s \{e(P_1, P_2)^{u\mu}\}$,
$C_1 = sP_1\{+\mu P_1\}$, $C_2 = sbP_1$, $C_3 = s(U_1 + \mathsf{id}V_1 + \mathsf{tag}W_1)\{+\mu(d + \mathsf{id} \cdot e + \mathsf{tag} \cdot c)P_1\}$.

Key: Contains five elements $(K_1, \ldots, K_5)$ defined as follows.

$r \xleftarrow{U} \mathbb{Z}_p$, $\{\gamma, \pi \xleftarrow{U} \mathbb{Z}_p\}$
$K_1 = rP_2$, $K_2 = rcP_2\{+\gamma P_2\}$, $K_3 = (u + r(d + \mathsf{id}e))\, P_2\{+\gamma\pi P_2\}$,
$K_4 = -r\Delta_3 P_2\{-\frac{\gamma}{b}P_2\}$, $K_5 = (-\Delta_4 - r(\Delta_1 + \mathsf{id}\Delta_2))\, P_2\{-\frac{\gamma\pi}{b}P_2\}$.

Note 1. In JR-IBE [28], b is mentioned to be an element of $\mathbb{Z}_p$. This is an oversight and b should be an element of $\mathbb{Z}_p^\times$ as we have mentioned above. This is because if b is zero, then division by b and consequently the definitions of the semi-functional components will not be meaningful.

The original JR-IBE scheme in [28] was proved to be secure based on the SXDH assumption. Straightforward modifications of proof in [28] will also show the security of the variant JR-IBE-D under the same assumption. For the sake of completeness, we state the security theorem JR-IBE-D.

Theorem 1. *If* $(\varepsilon_{\mathrm{DDH1}}, t_1)$-DDH1 *and* $(\varepsilon_{\mathrm{DDH2}}, t_2)$-DDH2 *assumptions hold in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *respectively, then* JR-IBE-D *is* (ε, t)-ANO-IND-ID-CPA*-secure where* $\varepsilon \leq \varepsilon_{\mathrm{DDH1}} + q \cdot \varepsilon_{\mathrm{DDH2}} + (q/p)$, $t_1 = t + O(\rho)$ *and* $t_2 = t + O(\rho)$. ρ *is the maximum time required for one scalar multiplication in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$.

4 Our CC-HIBE Constructions

Both schemes $\mathcal{H}_1$ and $\mathcal{H}_2$ are 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.

As is typical with BBG-type extensions the element V_1 is replaced with h elements $V_{1,1}, \ldots, V_{1,h}$ – one for each level of an identity. The set $U_1, (V_{1,j})_{j\in[1,h]}$ is used to create the identity hash – for an identity $\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell)$, the hash is given by $U_1 + \sum_{j=1}^{\ell} \mathsf{id}_j V_{1,j}$. Element W_1 will be retained to append the tag-component to the hash. This replaces the hash in JR-IBE-D ciphertext without affecting the number of elements in the ciphertext. Moreover, since the hash is embedded in a single ciphertext component, only one tag is required. Note that the keys in JR-IBE-D have two sub-hashes that when combined during decryption cancels with the hash of the ciphertext.

In JR-IBE-D, each of U_1, V_1, W_1 is split into two components kept as part of the master secret. The two sets of components determine the sub-hashes required in generating keys. Similarly, for the HIBE, we need to split $V_{1,j}$ for all $j \in [1, h]$ as $V_{1,j} = b\Delta_{2,j} + e_j$ where $\Delta_{1,j}, e_j \xleftarrow{\text{U}} \mathbb{Z}_p$. So the sub-hashes are determined by the vectors $\mathbf{v}_1 = (d, e_1, \ldots, e_h)$ and $\mathbf{v}_2 = (\Delta_1, \Delta_{2,1}, \ldots, \Delta_{2,h})$. Rerandomisation of keys during delegation can be done in two possible ways – make the encodings of vectors $\mathbf{v}_1, \mathbf{v}_2$ along with Δ_3, c in $\mathbb{G}_2$ public; or provide appropriately randomised copies of these elements in the key.

The second method retains the anonymity property leading to the scheme $\mathcal{H}_1$. This is because the vectors $\mathbf{v}_1, \mathbf{v}_2$ can be used to test whether a given ciphertext is encrypted to a particular identity or not. Keeping them secret naturally leads to anonymity. The former method leads to the scheme $\mathcal{H}_2$ that has shorter keys and faster algorithms compared to $\mathcal{H}_1$. But the efficiency comes at the cost of losing anonymity. Due to space constraints we only describe $\mathcal{H}_1$ and discuss its security. A description of $\mathcal{H}_2$ followed by an outline of its security is provided in Appendices B and D.

4.1 Scheme $\mathcal{H}_1$

We define $\mathcal{H}_1 = (\mathcal{H}_1.\mathsf{Setup}, \mathcal{H}_1.\mathsf{Encrypt}, \mathcal{H}_1.\mathsf{KeyGen}, \mathcal{H}_1.\mathsf{Delegate}, \mathcal{H}_1.\mathsf{Decrypt})$ where the algorithms are as follows.

$\mathcal{H}_1.\mathsf{Setup}(\kappa)$: Generate a Type-3 pairing $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, F_1, F_2)$ based on the security parameter κ. Compute parameters as follows.

$P_1 \xleftarrow{\text{U}} \mathbb{G}_1^\times,\ P_2 \xleftarrow{\text{U}} \mathbb{G}_2^\times$
$\Delta_1, \Delta_3, \Delta_4, c, d, u, (\Delta_{2,j}, e_j)_{j=1}^h \xleftarrow{\text{U}} \mathbb{Z}_p,\ b \xleftarrow{\text{U}} \mathbb{Z}_p^\times,$

$U_1 = (-\Delta_1 b + d)P_1,\ V_{1,j} = (-\Delta_{2,j} b + e_j)P_1$ for $j = 1, \ldots, h,\ W_1 = (-\Delta_3 b + c)P_1,$
$g_T = e(P_1, P_2)^{-\Delta_4 b + u},$

$\mathcal{PP} : (P_1, bP_1, U_1, (V_{1,j})_{j=1}^h, W_1, g_T)$
$\mathcal{MSK} : (P_2, cP_2, \Delta_1, \Delta_3, \Delta_4, d, u, (\Delta_{2,j}, e_j)_{j=1}^h)$

$\mathcal{H}_1.\mathsf{Encrypt}(\mathcal{PP}, M, \mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell))$: Pick $\mathsf{tag}, s \xleftarrow{\text{U}} \mathbb{Z}_p$ and set the ciphertext $\mathcal{C} = (C_0, C_1, C_2, C_3, \mathsf{tag})$ where

$C_0 = M \cdot (g_T)^s,\ C_1 = sP_1,\ C_2 = sbP_1,\ C_3 = s(U_1 + \sum_{j=1}^{\ell} \mathsf{id}_j V_{1,j} + \mathsf{tag} W_1).$

$\mathcal{H}_1.\mathsf{KeyGen}(\mathcal{MSK}, \mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell))$: Pick $r_1, r_2 \xleftarrow{\text{U}} \mathbb{Z}_p$ and compute the secret key $\mathcal{SK}_{\mathbf{id}} = (\mathcal{S}_1, \mathcal{S}_2)$ for $\mathbf{id}$, with $\mathcal{S}_1 = ((K_i)_{i\in[1,5]}, (D_{1,j}, E_{1,j})_{j\in[\ell+1,h]})$ and $\mathcal{S}_2 = ((J_i)_{i\in[1,5]}, (D_{2,j}, E_{2,j})_{j\in[\ell+1,h]})$ where

$K_1 = r_1 P_2,\ K_2 = r_1 c P_2,\ K_3 = \left(u + r_1(d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j)\right) P_2,$
$K_4 = -r_1 \Delta_3 P_2,\ K_5 = \left(-\Delta_4 - r_1(\Delta_1 + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j})\right) P_2,$
$D_{1,j} = r_1 e_j P_2,\ E_{1,j} = -r_1 \Delta_{2,j} P_2$ for $j = \ell+1, \ldots, h,$

$J_1 = r_2 P_2,\ J_2 = r_2 c P_2,\ J_3 = r_2 \left(d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j\right) P_2,$
$J_4 = -r_2 \Delta_3 P_2,\ J_5 = -r_2(\Delta_1 + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j}) P_2,$
$D_{2,j} = r_2 e_j P_2,\ E_{2,j} = -r_2 \Delta_{2,j} P_2$ for $j = \ell+1, \ldots, h$

$\mathcal{H}_1.\mathsf{Delegate}(\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell), \mathsf{id}_{\ell+1})$: Let $\mathbf{id} : \mathsf{id}_{\ell+1} = (\mathsf{id}_1, \ldots, \mathsf{id}_{\ell+1})$. $\mathcal{SK}_{\mathbf{id}:\mathsf{id}_{\ell+1}}$ is generated from $\mathcal{SK}_{\mathbf{id}}$ as follows.

$\tilde{r}_1, \tilde{r}_2 \xleftarrow{\mathrm{U}} \mathbb{Z}_p^{\times},$

$K_1 \leftarrow K_1 + \tilde{r}_1 J_1,\ K_2 \leftarrow K_2 + \tilde{r}_1 J_2,\ K_3 \leftarrow (K_3 + \mathsf{id}_{\ell+1} D_{1,\ell+1}) + \tilde{r}_1 (J_3 + \mathsf{id}_{\ell+1} D_{2,\ell+1}),$
$K_4 \leftarrow K_4 + \tilde{r}_1 J_4,\ K_5 \leftarrow (K_5 + \mathsf{id}_{\ell+1} E_{1,\ell+1}) + \tilde{r}_1 (J_5 + \mathsf{id}_{\ell+1} E_{2,\ell+1}),$
$D_{1,j} \leftarrow D_{1,j} + \tilde{r}_1 D_{2,j},\ E_{1,j} \leftarrow E_{1,j} + \tilde{r}_1 E_{2,j}$ for $j = \ell+2, \ldots, h,$

$J_1 \leftarrow \tilde{r}_2 J_1,\ J_2 \leftarrow \tilde{r}_2 J_2,\ J_3 \leftarrow \tilde{r}_2 (J_3 + \mathsf{id}_{\ell+1} D_{2,\ell+1}),$
$J_4 \leftarrow \tilde{r}_2 J_4,\ J_5 \leftarrow \tilde{r}_2 (J_5 + \mathsf{id}_{\ell+1} E_{2,\ell+1}),$
$D_{2,j} \leftarrow \tilde{r}_2 D_{2,j},\ E_{2,j} \leftarrow \tilde{r}_2 E_{2,j}$ for $j = \ell+2, \ldots, h,$

setting $r_1 \leftarrow r_1 + \tilde{r}_1 r_2$ and $r_2 \leftarrow \tilde{r}_2 r_2$. Note that the new values of r_1 and r_2 have uniform and independent distribution over $\mathbb{Z}_p$ given that $r_1, r_2 \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and $\tilde{r}_1, \tilde{r}_2 \xleftarrow{\mathrm{U}} \mathbb{Z}_p^{\times}$. Hence the distribution of $\mathcal{SK}_{\mathbf{id}:\mathsf{id}_{\ell+1}}$ is same as that of a freshly generated key for $\mathbf{id} : \mathsf{id}_{\ell+1}$ via the $\mathcal{H}_1.\mathsf{KeyGen}$ algorithm.

$\mathcal{H}_1.\mathsf{Decrypt}(\mathcal{C}, \mathcal{SK}_{\mathbf{id}})$: Return M' computed as: $M' = \dfrac{C_0 \cdot e(C_3, K_1)}{e(C_1, \mathsf{tag} K_2 + K_3) e(C_2, \mathsf{tag} K_4 + K_5)}$.

Correctness: For all messages M, for all $1 \leq \ell \leq h$, for all identities $\mathbf{id}$ of length ℓ, for all $\mathcal{C}$ and $\mathcal{SK}_{\mathbf{id}}$ such that $\mathcal{C} \leftarrow \mathcal{H}_1.\mathsf{Encrypt}(M, \mathbf{id})$, $\mathcal{SK}_{\mathbf{id}} \leftarrow \mathcal{H}_1.\mathsf{KeyGen}(\mathcal{MSK}, \mathbf{id})$ and $M' = \mathcal{H}_1.\mathsf{Decrypt}(\mathcal{C}, \mathcal{SK}_{\mathbf{id}})$, it holds that $M' = M$. The following computation substantiates this claim. Let $(\mathcal{C} = (C_0, C_1, C_2, C_3)) = \mathcal{H}_1.\mathsf{Encrypt}(M, \mathbf{id}; s)$ and $(\mathcal{SK}_{\mathbf{id}} = (\mathcal{S}_1, \mathcal{S}_2)) = \mathcal{H}_1.\mathsf{KeyGen}(\mathcal{MSK}, \mathbf{id}; r_1, r_2)$ with $\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell)$. We show the computation in steps. Let $h_1 = d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j + \mathsf{tag} \cdot c$ and $h_2 = \Delta_1 + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j} + \mathsf{tag} \cdot \Delta_3$.

$$
\begin{aligned}
e(C_1, \mathsf{tag} K_2 + K_3) &= e(sP_1, \mathsf{tag} \cdot r_1 c P_2 + (u + r_1(d + \textstyle\sum_{j=1}^{\ell} \mathsf{id}_j e_j)) P_2) \\
&= e(sP_1, uP_2 + r_1(d + \textstyle\sum_{j=1}^{\ell} \mathsf{id}_j e_j + \mathsf{tag} \cdot c) P_2) \\
&= e(P_1, P_2)^{su} e(P_1, P_2)^{r_1 s h_1} \\
e(C_2, \mathsf{tag} \cdot K_4 + K_5) &= e(sbP_1, -\mathsf{tag} \cdot r_1 \Delta_3 P_2 - (\Delta_4 + r_1(\Delta_1 + \textstyle\sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j})) P_2) \\
&= e(sbP_1, -\Delta_4 P_2 - r_1(\Delta_1 + \textstyle\sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j} + \mathsf{tag} \cdot \Delta_3) P_2) \\
&= e(P_1, P_2)^{-sb\Delta_4} e(P_1, P_2)^{-r_1 s b h_2} \\
e(C_3, K_1) &= e(s(U_1 + \textstyle\sum_{j=1}^{\ell} \mathsf{id}_j V_{1,j} + \mathsf{tag} \cdot W_1), r_1 P_2) \\
&= e((-\Delta_1 b + d) P_1 + \textstyle\sum_{j=1}^{\ell} \mathsf{id}_j (-\Delta_{2,j} b + e_j) P_1 + \mathsf{tag} \cdot (-\Delta_3 b + c) P_1, P_2)^{r_1 s} \\
&= e(-(\Delta_1 + \textstyle\sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j} + \mathsf{tag} \cdot \Delta_3) b P_1, P_2)^{r_1 s} e((d + \textstyle\sum_{j=1}^{\ell} \mathsf{id}_j e_j + \mathsf{tag} \cdot c) P_1, P_2)^{r_1 s} \\
&= e(P_1, P_2)^{-r_1 s b h_2} e(P_1, P_2)^{r_1 s h_1}
\end{aligned}
$$

Then, the message M' obtained after decryption is given by

$$\begin{aligned} M' &= \frac{C_0 \cdot e(C_3, K_1)}{e(C_1, \mathsf{tag} \cdot K_2 + K_3) e(C_2, \mathsf{tag} \cdot K_4 + K_5)} \\ &= \frac{M \cdot g_T^s \cdot e(P_1, P_2)^{-r_1 s b h_2} e(P_1, P_2)^{r_1 s h_1}}{e(P_1, P_2)^{su} e(P_1, P_2)^{r_1 s h_1} e(P_1, P_2)^{-sb\Delta_4} e(P_1, P_2)^{-r_1 s b h_2}} \\ &= \frac{M \cdot e(P_1, P_2)^{(-\Delta_4 b + u)s}}{e(P_1, P_2)^{s(-\Delta_4 b + u)}} \\ &= M, \end{aligned}$$

as required.

The above holds as well for all $\mathcal{SK}_{\mathbf{id}}$ derived from secret keys for higher level identities through the $\mathcal{H}_1$.Delegate algorithm. This is because a derived key have the same distribution as a key generated by a fresh call to the $\mathcal{H}_1$.KeyGen algorithm which has been pointed out in the description of the $\mathcal{H}_1$.Delegate algorithm.

From a dual system perspective. One can see in Section 5 that the scalar u, along with scalars $d, c, e_{j\,j\in[1,h]}$, define the semi-functional ciphertext space for $\mathcal{H}_1$. These scalars provide the secret information for simulating semi-functional components. A crucial requirement for a dual system proof is that these scalars are statistically hidden from the adversary. Observe that the element g_T in the public parameters, information theoretically hides the element u. Similarly, elements $U_1, V_{1,j}, W_1$ hide the scalars d, e_j, c respectively. Further intuition with respect to the dual-system proof and a sketch of how the various scalars interact is provided in Section 5.

4.2 Anonymity and Constant-Size Ciphertexts

As mentioned in Section 1, many HIBE schemes (such as [25]) have the prefix decryption property. That is, an entity with identity $\mathbf{id}'$ and a corresponding secret key $\mathcal{SK}_{\mathbf{id}'}$ can decrypt any ciphertext corresponding to $\mathbf{id}$ where $\mathbf{id}'$ is a prefix of $\mathbf{id}$. As an example, consider the Gentry-Silverberg HIBE [25] (GS-HIBE) based on a symmetric pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. The ciphertext for an identity $\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell)$ consists of points $rP_0, rP_2, \ldots, rP_\ell$ from $\mathbb{G}$ and the n-bit string $M \oplus H_2(e(P_1, Q_0)^r)$ where $H_1 : \{0,1\}^* \to \mathbb{G}$, $H_2 : \mathbb{G}_T \to \{0,1\}^n$ are crytographic hash functions, $P_i = H_1(\mathsf{id}_1, \ldots, \mathsf{id}_i)$ for $i \in [1, \ell]$, elements P_0, Q_0 come from $\mathcal{PP}$ and M is the message. Note that when we remove the points $rP_{\ell'+1}, \ldots, rP_\ell$ ($\ell' < \ell$), the remaining components form a valid ciphertext for the identity $\mathbf{id}' = (\mathsf{id}_1, \ldots, \mathsf{id}_{\ell'})$ and hence can be decrypted using a secret key for $\mathbf{id}'$. The prefix decryption property holds in this case.

In the non-anonymous setting, $\mathbf{id}$ is known. Hence, decryption can be done via a secret key for $\mathbf{id}$ that is derived from $\mathcal{SK}_{\mathbf{id}'}$ by a sequence of calls to the Delegate algorithm. In case of anonymous HIBE, $\mathbf{id}$ is not known and as a result, delegation is not possible. If the ciphertext contains separate components corresponding to each level of $\mathbf{id}$, prefix decryption would still hold. As shown above (for the case of GS-HIBE), one can truncate the ciphertext retaining only the components corresponding to $\mathbf{id}'$ and then perform decryption using $\mathcal{SK}_{\mathbf{id}'}$. The anonymous HIBE of Okamoto-Takashima[35] also has this property. But in case of an anonymous HIBE where the ciphertext size is constant, as in $\mathcal{H}_1$, it is not possible to decrypt the ciphertext with $\mathcal{SK}_{\mathbf{id}'}$. The reason is that is no way to remove (or truncate) the randomised components corresponding $\mathbf{id} \setminus \mathbf{id}'$ from the ciphertext (here, $\mathbf{id} \setminus \mathbf{id}'$ denotes the suffix of $\mathbf{id}'$ in $\mathbf{id}$ i.e., $\mathsf{id}_{\ell'}, \ldots, \mathsf{id}_\ell$). More percisely, given the hash $s(U_1 + \sum_{j=1}^{\ell} \mathsf{id}_j V_{1,j} + \mathsf{tag} W_1)$ for $\mathbf{id}$ it is impossible to extract a hash for $\mathbf{id}'$ since we have no knowledge of $sV_{1,j}$'s. This limitation is not particular to our work and neither

is something that arises due to the techniques that we use. It is an inherent limitation and is present in all previously known HIBE constructions which simultaneously achieve constant-size ciphertexts and anonymity.

On the contrary, as pointed out in Section 1, the absence of prefix decryption property is acceptable and possibly useful in certain applications. For related discussions on this issue, the reader is refered to [25] and [17].

5 Security of $\mathcal{H}_1$

The scheme $\mathcal{H}_1$ is proved secure in the sense of ANO-IND-ID-CPA (described in Section A) following the dual system methodology introduced by Waters [46]. We first provide algorithms $\mathcal{H}_1$.SFEncrypt and $\mathcal{H}_1$.SFKeyGen that generate semi-functional ciphertexts and keys (respectively) required for a dual system proof. In addition, we need an algorithm PSFKeyGen that generates *partial semi-functional keys* ([40]). These are required only in the security proof of $\mathcal{H}_1$ and not $\mathcal{H}_2$.

$\mathcal{H}_1$.SFEncrypt$(\mathcal{MSK}, \mathcal{C})$: Let $(\mathcal{C} = (C_0, C_1, C_2, C_3)) \leftarrow \mathcal{H}_1$.Encrypt$(m, \mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell))$. Pick $\mu \xleftarrow{\text{U}} \mathbb{Z}_p$ and modify the components of $\mathcal{C}$ as follows.

$$C_0 \leftarrow C_0 \cdot e(P_1, P_2)^{u\mu}, \quad C_1 \leftarrow C_1 + \mu P_1, \quad C_2 \leftarrow C_2, \quad C_3 \leftarrow C_3 + \mu(d + \textstyle\sum_{j=1}^{\ell} \mathsf{id}_j e_j + \mathsf{tag} \cdot c)P_1.$$

Return the modified ciphertext $\mathcal{C} = (C_0, C_1, C_2, C_3)$.

$\mathcal{H}_1$.SFKeyGen$(\mathcal{MSK}, \mathcal{SK}_{\mathbf{id}})$: This algorithm takes in a normal secret key $\mathcal{SK}_{\mathbf{id}} = (\mathcal{S}_1, \mathcal{S}_2)$ for identity $\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell)$ and generates a semi-functional key as follows.

$\gamma_1, \gamma_2, \pi, \sigma, (\pi_j, \sigma_j)_{j=1}^{h} \xleftarrow{\text{U}} \mathbb{Z}_p$,

$$K_1 \leftarrow K_1, \;\; K_2 \leftarrow K_2 + \gamma_1 P_2, \;\; K_3 \leftarrow K_3 + \gamma_1 \pi P_2, \;\; K_4 \leftarrow K_4 - \left(\frac{\gamma_1}{b}\right) P_2, \;\; K_5 \leftarrow K_5 - \left(\frac{\gamma_1 \pi}{b}\right) P_2,$$
$$D_{1,j} \leftarrow D_{1,j} + \gamma_1 \pi_j P_2, \;\; E_{1,j} \leftarrow E_{1,j} - \left(\frac{\gamma_1 \pi_j}{b}\right) P_2 \quad \text{for } j = \ell + 1, \ldots, h,$$

$$J_1 \leftarrow J_1, \;\; J_2 \leftarrow J_2 + \gamma_2 P_2, \;\; J_3 \leftarrow J_3 + \gamma_2 \sigma P_2, \;\; J_4 \leftarrow J_4 - \left(\frac{\gamma_2}{b}\right) P_2, \;\; J_5 \leftarrow J_5 - \left(\frac{\gamma_2 \sigma}{b}\right) P_2,$$
$$D_{2,j} \leftarrow D_{2,j} + \gamma_2 \sigma_j P_2, \;\; E_{2,j} \leftarrow E_{2,j} - \left(\frac{\gamma_2 \sigma_j}{b}\right) P_2 \quad \text{for } j = \ell + 1, \ldots, h,$$

The resulting key $\mathcal{SK}_{\mathbf{id}} = (\mathcal{S}_1, \mathcal{S}_2)$ is returned.

PSFKeyGen$(\mathcal{MSK}, \mathcal{SK}_{\mathbf{id}})$: Returns a key $\mathcal{SK}_{\mathbf{id}}$ for identity $\mathbf{id}$ with $\mathcal{S}_1$-components having semi-functional terms (generated according to $\mathcal{H}_1$.SFKeyGen algorithm) and $\mathcal{S}_2$-components being normal (as returned by $\mathcal{H}_1$.KeyGen algorithm).

Discussion. It is natural to ask whether it is at all required to define semi-functional terms for $\mathcal{S}_2$ components of a key that do not play any role in decryption. The answer is yes and the reason is as follows. Since all the elements required to create the id-hash in $\mathbb{G}_2$ are hidden, there is no way to test the identity to which a ciphertext is encrypted. The scheme seems to be anonymous but to prove it, we need to ensure that a semi-functional encryption to a target identity is indistinguishable from a semi-functional encryption to a random identity vector. (We need semi-functionality in order to deal with the key extraction queries.)

Normally, the K-components of the key are used for decrypting a ciphertext. When these are paired with the ciphertext components we obtain the blinding factor for the message that only depends on Δ_4, u and the randomiser s. Instead if we try decrypting using J-components of the key (which do not have Δ_4 and u terms), we get 1_T, the identity of $\mathbb{G}_T$. Hence the J-components help in testing whether the

ciphertext is indeed encrypted under **id** or not. The presence of such a test does not help in proving anonymity property. Therefore, it is essential to make $\mathcal{S}_2$-components of all keys semi-functional before arguing about anonymity.

It is straightforward to see that decryption of a semi-functional ciphertext by a normal key or that of a normal ciphertext with a semi-functional key succeeds. When both ciphertext and key are semi-functional, decryption results in an extra masking factor of $e(P_1, P_2)^{\gamma\mu(\mathsf{tag}+\pi)}$ on the message. Decryption is only successful if $\pi = -\mathsf{tag}$ whence the ciphertext and key become *nominally semi-functional*.

The following theorem states precisely the security guarantee we obtain for $\mathcal{H}_1$.

Theorem 2. *If* $(\varepsilon_{\text{DDH1}}, t_1)$-DDH1 *and* $(\varepsilon_{\text{DDH2}}, t_2)$-DDH2 *assumptions hold in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *respectively, then* $\mathcal{H}_1$ *is* (ε, t)-ANO-IND-ID-CPA*-secure where* $\varepsilon \leq \varepsilon_{\text{DDH1}} + 2q \cdot \varepsilon_{\text{DDH2}} + (2q/p)$, $t_1 = t + O(h\rho)$ *and* $t_2 = t + O(h\rho)$. ρ *is the maximum time required for one scalar multiplication in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$.

Proof Sketch. Fix any t-time adversary $\mathscr{A}$. Let G_{real} denote the HIBE security game ano-ind-cpa (described in Section A) and G_{final} be a game where all keys are semi-functional and the challenge ciphertext is a semi-functional encryption of a random message to a random identity vector. The probability that $\mathscr{A}$ wins in G_{final} is 1/2. To prove the theorem, we need to show a bound on $\mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}cpa}}_{\mathcal{H}_1}(\mathscr{A}) = |\Pr[\mathscr{A} \text{ wins in } \mathsf{G}_{real}] - (1/2)|$ which is equivalent to bounding $|\Pr[\mathscr{A} \text{ wins in } \mathsf{G}_{real}] - \Pr[\mathscr{A} \text{ wins in } \mathsf{G}_{final}]|$. In order to obtain this bound, we first define a sequence of games starting from G_{real} and making small changes until we reach G_{final}. Define $\mathsf{G}_{k,0}$, $1 \leq k \leq q$ similar to G_{real} except that challenge ciphertext is semi-functional, first $k-1$ keys are semi-functional and k-th key is partial semi-functional. In $\mathsf{G}_{k,1}$, $0 \leq k \leq q$, the challenge ciphertext is semi-functional and first k keys are semi-functional. The game sequence is G_{real}, $\mathsf{G}_{0,1}$, $(\mathsf{G}_{k,0}, \mathsf{G}_{k,1})_{k=1}^{q}$, G_{final}. The advantage of $\mathscr{A}$ in winning G_{real} can now be bounded in terms of its advantage in distinguishing between successive games. This is done via reductions from the SXDH problem to the task of distinguishing between successive games. Essentially, there are two kinds of reductions - first and second. In the first reduction, we show that $\mathscr{A}$'s ability to distinguish between G_{real} and $\mathsf{G}_{0,1}$ can be used to solve a DDH1 instance. The second reduction shows that an algorithm $\mathscr{A}$ that can distinguish between $\mathsf{G}_{k-1,1}$ and $\mathsf{G}_{k,0}$ for some $k \in [1, q]$, can be used to construct an algorithm $\mathscr{B}_2$ solving DDH2. Similar arguments hold for all values of k and also for the transition from $\mathsf{G}_{k,0}$ to $\mathsf{G}_{k,1}$. The final transition i.e, $\mathsf{G}_{q,1}$ to G_{final} is done just by changing the way information provided to $\mathscr{A}$ is generated so that the distribution of $\mathscr{A}$'s view in the two games are statistically indistinguishable except with probability $2q/p$.. We now provide an outline of each stage in the proof.

First Reduction: Suppose that $\mathscr{B}_1$ is a DDH1-solver. $\mathscr{B}_1$ simulates the game using a DDH1 instance $(\mathcal{G}, P_1, bP_1, sbP_1, P_2, (s+\mu)P_1)$. The element b of the instance corresponds to the scalar b of the scheme. $\mathscr{B}_1$ sets up the system normally since it has all information required to do so. The master secret is also known since none of its components depend on b. Furthermore, it cannot create semi-functional keys as no encoding of b in $\mathbb{G}_2$ is provided. All the key extract queries are answered normally. $\mathscr{B}_1$ sets the randomiser for the challenge ciphertext $\widehat{\mathcal{C}}$ to be s (from the instance). $\widehat{\mathcal{C}}$ will be normal or semi-functional depending on whether the instance is real i.e., $\mu = 0$, or random ($\mu \xleftarrow{\text{U}} \mathbb{Z}_p$).

Second Reduction: The DDH2-solver $\mathscr{B}_2$ obtains an instance $(\mathcal{G}, P_1, P_2, rP_2, cP_2, (rc+\gamma)P_2)$. Here c corresponds to the scalar c in $\mathcal{MSK}$. Elements $d, (e_j)_{j\in[1,h]}$ are set to random degree-1 polynomials in c. Scalar b is chosen randomly from $\mathbb{Z}_p^{\times}$. Let $\mathbf{y} = (d, e_1, \ldots, e_h)$. The public parameters are created differently since $\mathbf{y}$ is not known. Only its encoding in $\mathbb{G}_2$ i.e, $\mathbf{y}P_2$ is known. Specifically $U_1, V_{1,j}, W_1$ are chosen at random from $\mathbb{G}_1$. Depending on these and $\mathbf{y}$, the corresponding Δ's are implicitly set. Encodings of Δ's can be computed only in $\mathbb{G}_2$. This enables normal key generation as well as semi-functional key generation. In its response to the k-th key extract query, $\mathscr{B}_2$ maps r from the instance to

the randomiser r_1 in the key. Accordingly it generates the key choosing r_2 at random. If $\gamma = 0$, the key will be normal. Otherwise the key is partial semi-functional and γ corresponds to the randomiser γ_1 in the semi-functional part. Moreover, a linear polynomial $f(\mathbf{id}_k)$ in $\mathbf{id}_k$-components is embedded in the semi-functional scalar π. This polynomial is determined by the co-efficients of c in $\mathbf{y}$. The coefficients of c in e_j also determine π_j respectively. For the challenge ciphertext, $\mathscr{B}_2$ has to create semi-functional components which depend on $\mathbf{y}$. But $\mathbf{y}$ depends on c and encoding of c in $\mathbb{G}_1$ is not known. The only way out is to set $\mathsf{tag} = -f(\widehat{\mathbf{id}}_\beta)$ so that terms depending on c vanish. A consequence is that $\mathscr{B}_2$ can only generate nominally semi-functional ciphertext for $\mathbf{id}_k$. We then argue that the simulation is perfect.

Final Transition: It is required to show that $\mathsf{G}_{q,1}$ and G_{final} are statistically indistinguishable from the attacker's point of view except for probability at most $2q/p$. The generation of public parameters and keys provided to $\mathscr{A}$ are changed ensuring that their form is equivalent to that in $\mathsf{G}_{q,1}$ and they are independent of the scalars $u, d, (e_j)_{j\in[1,h]}$. Consequently the challenge ciphertext is the only place where these scalars come into play, especially in those components that consist of the identity-hash and the message. Basically, the message and the id-hash are masked by random quantities so that G_{final} is simulated.

Refer to Appendix C for details of the proof.

6 Conclusion

We obtain two HIBE schemes with constant-size ciphertexts and full security from the IBE scheme of Jutla and Roy. One achieves anonymity while the other is non-anonymous with shorter keys. Compared to previous HIBE schemes, our constructions provide very good efficiency with just 3 pairings for decryption and 3 group elements in the ciphertext. These are also the only CC-HIBEs achieving security under standard assumptions and degradation independent of the HIBE depth. In HIBE-related literature focussed on either constant-size ciphertexts or anonymity or both, we believe that our constructions complete the picture.

Acknowledgement

We thank reviewers of a previous version of this paper 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. Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT*, volume 8441 of *Lecture Notes in Computer Science*, pages 557–577. Springer, 2014.
3. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.
4. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity-based encryption with constant size ciphertext. In Cramer [19], pages 440–456. Full version available at Cryptology ePrint Archive; Report 2005/015.
5. 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.
6. 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.

7. Ran Canetti and Juan A. Garay, editors. *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*. Springer, 2013.
8. Sanjit Chatterjee, Darrel Hankerson, Edward Knapp, and Alfred Menezes. Comparing two pairing-based aggregate signature schemes. *Des. Codes Cryptography*, 55(2-3):141–167, 2010.
9. Sanjit Chatterjee and Alfred Menezes. On cryptographic protocols employing asymmetric pairings – the role of ψ revisited. *Discrete Applied Mathematics*, 159(13):1311–1322, 2011.
10. Sanjit Chatterjee and Palash Sarkar. HIBE with short public parameters without random oracle. In X. Lai and K. Chen, editors, *ASIACRYPT*, volume 4284 of *Lecture Notes in Computer Science*, pages 145–160. Springer, 2006. see also Cryptology ePrint Archive, Report 2006/279, http://eprint.iacr.org/.
11. Sanjit Chatterjee and Palash Sarkar. New constructions of constant size ciphertext HIBE without random oracle. In M.S. Rhee and B. Lee, editors, *ICISC*, volume 4296 of *Lecture Notes in Computer Science*, pages 310–327. Springer, 2006.
12. Sanjit Chatterjee and Palash Sarkar. Constant size ciphertext HIBE in the augmented selective-id model and its extensions. *J. UCS*, 13(10):1367–1395, 2007.
13. Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Hoeteck Wee. Shorter IBE and signatures via asymmetric pairings. *IACR Cryptology ePrint Archive*, 2012:224, 2012.
14. Jie Chen and Hoeteck Wee. Fully, (almost) tightly secure IBE and dual system groups. `https://sites.google.com/site/jchencrypto/publications`, 2013.
15. Jie Chen and Hoeteck Wee. Fully, (almost) tightly secure IBE and dual system groups. In Canetti and Garay [7], pages 435–460. Full version available as IACR Technical Report, 2013/803, `http://eprint.iacr.org/2013/803`.
16. Jie Chen and Hoeteck Wee. Dual system groups and its applications — compact hibe and more. Cryptology ePrint Archive, Report 2014/265, 2014. `http://eprint.iacr.org/`.
17. Sherman S. M. Chow. Removing Escrow from Identity-Based Encryption. In Jarecki and Tsudik [27], pages 256–276.
18. 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.
19. Ronald Cramer, editor. *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*. Springer, 2005.
20. 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.
21. 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.
22. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge L. Villar. An algebraic framework for diffie-hellman assumptions. In Canetti and Garay [7], pages 129–147.
23. Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.
24. Craig Gentry and Shai Halevi. Hierarchical identity based encryption with polynomially many levels. In Omer Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 437–456. Springer, 2009.
25. 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.
26. 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.
27. Stanislaw Jarecki and Gene Tsudik, editors. *Public Key Cryptography - PKC 2009, 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings*, volume 5443 of *Lecture Notes in Computer Science*. Springer, 2009.
28. Charanjit S. Jutla and Arnab Roy. Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT (1)*, volume 8269 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2013.
29. 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.
30. 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.
31. Allison B. Lewko. Tools for Simulating Features of Composite Order Bilinear Groups in the Prime Order Setting. In Pointcheval and Johansson [38], pages 318–335.
32. 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.
33. Tatsuaki Okamoto and Katsuyuki Takashima. Homomorphic Encryption and Signatures from Vector Decomposition. In Steven D. Galbraith and Kenneth G. Paterson, editors, *Pairing*, volume 5209 of *Lecture Notes in Computer Science*, pages 57–74. Springer, 2008.

34. Tatsuaki Okamoto and Katsuyuki Takashima. Hierarchical Predicate Encryption for Inner-Products. In Mitsuru Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 214–231. Springer, 2009.
35. Tatsuaki Okamoto and Katsuyuki Takashima. Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption. In Dongdai Lin, Gene Tsudik, and Xiaoyun Wang, editors, *CANS*, volume 7092 of *Lecture Notes in Computer Science*, pages 138–159. Springer, 2011.
36. Tatsuaki Okamoto and Katsuyuki Takashima. Adaptively Attribute-Hiding (Hierarchical) Inner Product Encryption. In Pointcheval and Johansson [38], pages 591–608.
37. 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.
38. David Pointcheval and Thomas Johansson, editors. *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*. Springer, 2012.
39. Somindu C. Ramanna, Sanjit Chatterjee, and Palash Sarkar. Variants of waters' dual system primitives using asymmetric pairings - (extended abstract). In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 298–315. Springer, 2012.
40. Somindu C. Ramanna and Palash Sarkar. Anonymous constant-size ciphertext HIBE from asymmetric pairings. In Martijn Stam, editor, *IMA Int. Conf.*, volume 8308 of *Lecture Notes in Computer Science*, pages 344–363. Springer, 2013.
41. Jae Hong Seo, Tetsutaro Kobayashi, Miyako Ohkubo, and Koutarou Suzuki. Anonymous hierarchical identity-based encryption with constant size ciphertexts. In Jarecki and Tsudik [27], pages 215–234.
42. 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.
43. Elaine Shi and Brent Waters. Delegating capabilities in predicate encryption systems. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (2)*, volume 5126 of *Lecture Notes in Computer Science*, pages 560–578. Springer, 2008.
44. Nigel P. Smart and Frederik Vercauteren. On computable isomorphisms in efficient asymmetric pairing-based systems. *Discrete Applied Mathematics*, 155(4):538–547, 2007.
45. Brent Waters. Efficient identity-based encryption without random oracles. In Cramer [19], pages 114–127.
46. 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.
47. Hoeteck Wee. Dual system encryption via predicate encodings. In Yehuda Lindell, editor, *TCC*, volume 8349 of *Lecture Notes in Computer Science*, pages 616–637. Springer, 2014.

A Hierarchical Identity-Based Encryption

Definition. 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 input a security parameter κ and maximum depth of the HIBE h, this algorithm 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.

Correctness. The HIBE scheme is said to satisfy the correctness condition if for all $1 \leq \ell \leq h$, for all identities $\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell)$, for all $\mathcal{SK}_{\mathbf{id}} \leftarrow \mathsf{KeyGen}(\mathcal{MSK}, \mathbf{id})$ or $\mathcal{SK}_{\mathbf{id}} \leftarrow \mathsf{Delegate}(\mathbf{id} \setminus \mathsf{id}_\ell, \mathsf{id}_\ell)$, for all messages M and for all $\mathcal{C} \longleftarrow \mathsf{Encrypt}(\mathbf{id}, M)$, the output of $\mathsf{Decrypt}(\mathcal{C}, \mathbf{id}, \mathcal{SK}_{\mathbf{id}})$ equals M with probability 1.

Security Model. The standard notion of security of a HIBE scheme is indistinguishability of ciphertexts against a chosen plaintext attack (first described by [25]). It is modelled by the following security game, called ind-cpa.

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 M_0, M_1 and identity $\widehat{\mathbf{id}}$ as challenge with the restriction that no prefix of $\widehat{\mathbf{id}}$ has 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 $\widehat{\mathbf{id}}$ 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 $\widehat{\mathbf{id}}$.

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 ind-cpa given by

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

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

Anonymity. The security game defined below captures both anonymity and security against a chosen plaintext attack for HIBE schemes. This model, which we call ano-ind-cpa, is equivalent to the standard security notions for IND-ID-CPA-security and anonymity taken together and has been used earlier in [21, 20].

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 ano-ind-cpa given by

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

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

B Scheme $\mathcal{H}_2$

This section presents the second (non-anonymous) HIBE construction. As discussed in Section 4, two sub-hashes in the key are combined to form the identity-hash required for cancellation with the ciphertext. The sub-hashes are determined by the vectors $\mathbf{v}_1 = (d, e_1, \ldots, e_h)$ and $\mathbf{v}_2 = (\Delta_1, \Delta_{2,1}, \ldots, \Delta_{2,h})$. In order to realise anonymity, these vectors are kept as part of the master secret in $\mathcal{H}_1$. Additional elements had to be provided in the key to enable rerandomisation during delegation. It turns out that we can obtain a non-anonymous scheme by making these vectors public. The availablity of these vectors facilitates rerandomisation and hence the keys no longer need extra components for this purpose. As a result, keys are shorter and algorithms KeyGen, $\mathsf{Delegate}$ are more efficient in comparison to $\mathcal{H}_1$.

The method of followed here in obtaining a non-anonymous HIBE did not work out for previously known anonymous HIBE schemes [40, 30]. This is due to the following reasons. The element $\mathbb{G}_T$ would be of the form $e(P_1, P_2)^\alpha$ where α is part of the master secret. P_1 and P_2 would be required for encryption and delegation respectively as a result of which both P_1 and P_2 would be present in $\mathcal{PP}$. However, this leaks α information theoretically thus revealing the message too! The splitting of α here in terms of Δ_4 and u precisely overcomes this problem. These scalars further provide the randomness required to generate semi-functional components.

Regarding the dual system proof, we mentioned in Section 4 that some elements in the master secret provide the randomness required to generate semi-functional components during simulation. In $\mathcal{H}_2$, the scalars $d, c, (e_j)_{j\in[1,h]}$ are revealed information theoretically in the public parameters. Although d, c, e_j being hidden provides more randomness, they are not essential to generating the required amount of randomness in the proof. The scalar u, hidden by $\mathbb{G}_T$ in the public parameters, is sufficient.

We define $\mathcal{H}_2 = (\mathcal{H}_2.\mathsf{Setup}, \mathcal{H}_2.\mathsf{Encrypt}, \mathcal{H}_2.\mathsf{KeyGen}, \mathcal{H}_2.\mathsf{Delegate}, \mathcal{H}_2.\mathsf{Decrypt})$ where the algorithms are as follows.

$\mathcal{H}_2.\mathsf{Setup}(\kappa)$: Generate a Type-3 pairing $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, F_1, F_2)$ based on the security parameter κ. Compute parameters as follows.

$P_1 \xleftarrow{\text{U}} \mathbb{G}_1^\times,\ P_2 \xleftarrow{\text{U}} \mathbb{G}_2^\times$
$\Delta_1, \Delta_3, \Delta_4, c, d, u, (\Delta_{2,j}, e_j)_{j=1}^h \xleftarrow{\text{U}} \mathbb{Z}_p,\ b \xleftarrow{\text{U}} \mathbb{Z}_p^\times,$

$U_1 = (-\Delta_1 b + d)P_1,\ V_{1,j} = (-\Delta_{2,j} b + e_j)P_1$ for $j = 1, \ldots, h,\ W_1 = (-\Delta_3 b + c)P_1,$
$g_T = e(P_1, P_2)^{-\Delta_4 b + u},$

$\mathcal{PP} : (P_1, bP_1, U_1, (V_{1,j})_{j=1}^h, W_1, P_2, \Delta_1 P_2, \Delta_3 P_2, dP_2, cP_2, (\Delta_{2,j} P_2, e_j P_2)_{j=1}^h, g_T)$
$\mathcal{MSK} : (\Delta_4, u)$

$\mathcal{H}_2.\mathsf{Encrypt}(\mathcal{PP}, M, \mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell))$: Pick $\mathsf{tag}, s \xleftarrow{\text{U}} \mathbb{Z}_p$ and set the ciphertext $\mathcal{C} = (C_0, C_1, C_2, C_3, \mathsf{tag})$ where

$C_0 = M \cdot (g_T)^s,\ C_1 = sP_1,\ C_2 = sbP_1,\ C_3 = s(U_1 + \sum_{j=1}^{\ell} \mathsf{id}_j V_{1,j} + \mathsf{tag} W_1).$

$\mathcal{H}_2.\mathsf{KeyGen}(\mathcal{MSK}, \mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell))$: Pick $r \xleftarrow{\text{U}} \mathbb{Z}_p$ and compute the secret key $\mathcal{SK}_{\mathbf{id}} = ((K_i)_{i\in[1,5]}, (D_j, E_j)_{j\in[\ell+1,h]})$ for $\mathbf{id}$ where,

$K_1 = rP_2,\ K_2 = rcP_2,\ K_3 = \left(u + r(d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j)\right) P_2,$
$K_4 = -r\Delta_3 P_2,\ K_5 = \left(-\Delta_4 - r(\Delta_1 + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j})\right) P_2,$
$D_j = re_j P_2,\ E_j = -r\Delta_{2,j} P_2$ for $j = \ell + 1, \ldots, h.$

$\mathcal{H}_2$.Delegate($\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell), \mathsf{id}_{\ell+1}$): Let $\mathbf{id} : \mathsf{id}_{\ell+1} = (\mathsf{id}_1, \ldots, \mathsf{id}_{\ell+1})$. $\mathcal{SK}_{\mathbf{id}:\mathsf{id}_{\ell+1}}$ is generated from $\mathcal{SK}_{\mathbf{id}}$ as follows.

$\tilde{r} \xleftarrow{\mathrm{U}} \mathbb{Z}_p^\times$,

$K_1 \leftarrow K_1 + \tilde{r}P_2,\ K_2 \leftarrow K_2 + \tilde{r}cP_2,\ K_3 \leftarrow (K_3 + \mathsf{id}_{\ell+1}D_{\ell+1}) + \tilde{r}(d + \sum_{j=1}^{\ell+1} \mathsf{id}_j e_j)P_2,$
$K_4 \leftarrow K_4 - \tilde{r}\Delta_3 P_2,\ K_5 \leftarrow (K_5 + \mathsf{id}_{\ell+1}E_{\ell+1}) - \tilde{r}(\Delta_1 + \sum_{j=1}^{\ell+1} \mathsf{id}_j \Delta_{2,j})P_2,$
$D_j \leftarrow D_j + \tilde{r}e_j P_2,\ E_j \leftarrow E_j - \tilde{r}\Delta_{2,j}P_2$ for $j = \ell+2, \ldots, h,$

setting $r \leftarrow r + \tilde{r}$. Note that the distribution of $\mathcal{SK}_{\mathbf{id}:\mathsf{id}_{\ell+1}}$ is same as that of a freshly generated key for $\mathbf{id} : \mathsf{id}_{\ell+1}$ via the KeyGen algorithm.

$\mathcal{H}_2$.Decrypt($\mathcal{C}, \mathcal{SK}_{\mathbf{id}}$): Return M' computed as:

$$M' = \frac{C_0 \cdot e(C_3, K_1)}{e(C_1, \mathsf{tag}K_2 + K_3)e(C_2, \mathsf{tag}K_4 + K_5)}.$$

Note 2. The encryption and decryption algorithms of $\mathcal{H}_1$ and $\mathcal{H}_2$ are identical and hence the correctness of decryption for $\mathcal{H}_2$ follows from that of $\mathcal{H}_1$.

Note 3. The KeyGen and Delegate algorithms for $\mathcal{H}_2$ are identical to the portion of the corresponding algorithms for $\mathcal{H}_1$ which modify the $\mathcal{S}_1$-components of the key. The $\mathcal{S}_2$ components of the key in $\mathcal{H}_1$ are not required in $\mathcal{H}_2$.

Discussion. Setting $h = 1$ in $\mathcal{H}_2$ yields a non-anonymous variant of JR-IBE-D. The resulting IBE has efficiency comparable to JR-IBE-D but has seven extra elements from $\mathbb{G}_2$ in public parameters. It is interesting to note that $\mathcal{H}_2$ is the only known HIBE within the dual system framework which has rerandomisable keys. The same holds for the corresponding IBE as well.

C Proof of Theorem 2

Consider a sequence of games G_{real}, $\mathsf{G}_{0,1}$, $(\mathsf{G}_{k,0}, \mathsf{G}_{k,1})_{k=1}^{q}$, G_{final} between an adversary $\mathscr{A}$ and a challenger with the games defined as follows.

- G_{real}: the actual HIBE security game ano-ind-cpa (described in Section A).
- $\mathsf{G}_{k,0}$, $1 \le k \le q$: challenge ciphertext is semi-functional; first $k-1$ keys are semi-functional and k-th key is partial semi-functional.
- $\mathsf{G}_{k,1}$, $0 \le k \le q$: challenge ciphertext is semi-functional; first k keys are semi-functional.
- G_{final}: challenge ciphertext is a semi-functional encryption of a random message under a random identity vector; all keys are semi-functional.

Let $X_\square$ denote the event that $\mathscr{A}$ wins in $\mathsf{G}_\square$. Clearly, the bit β is statistically hidden from the attacker in G_{final}, which means that $\Pr[X_{final}] = 1/2$.

In Lemmas 1, 2, 3 and 4, we show that

- $|\Pr[X_{real}] - \Pr[X_{0,1}]| \le \varepsilon_{\mathrm{DDH1}}$,
- $|\Pr[X_{k-1,1}] - \Pr[X_{k,0}]| \le \varepsilon_{\mathrm{DDH2}}$,
- $|\Pr[X_{k,0}] - \Pr[X_{k,1}]| \le \varepsilon_{\mathrm{DDH2}}$,
- $|\Pr[X_{q,1}] - \Pr[X_{final}]| \le 2q/p$.

The advantage of $\mathscr{A}$ in breaking the security of $\mathcal{H}_1$ is thus given by

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}cpa}}_{\mathcal{H}_1}(\mathscr{A}) &= |\Pr[X_{real}] - \frac{1}{2}| \\
&= |\Pr[X_{real}] - \Pr[X_{final}]| \\
&\leq |\Pr[X_{real}] - \Pr[X_{0,1}]| + \sum_{k=1}^{q}(|\Pr[X_{k-1,1}] - \Pr[X_{k,0}]| + |\Pr[X_{k,0}] - \Pr[X_{k,1}]|) \\
&\quad + |\Pr[X_{q,1}] - \Pr[X_{final}]| \\
&\leq \varepsilon_{\text{DDH1}} + 2q\varepsilon_{\text{DDH2}} + \frac{2q}{p}.
\end{aligned}$$

In the sequel, $\mathscr{B}_1$ (resp. $\mathscr{B}_2$) is a DDH1-solver (resp. DDH2-solver). We argue that $\mathscr{B}_1$, using the adversary's ability to distinguish between G_{real} and $\mathsf{G}_{0,1}$, can solve DDH1. Similarly, $\mathscr{A}$'s power to distinguish between $\mathsf{G}_{k-1,1}$ and $\mathsf{G}_{k,0}$ (or $\mathsf{G}_{k,0}$ and $\mathsf{G}_{k,1}$) for $k \in [1, q]$, can be leveraged to build a DDH2-solver $\mathscr{B}_2$.

Lemma 1. $|\Pr[X_{real}] - \Pr[X_{0,1}]| \leq \varepsilon_{\text{DDH1}}$.

Proof. Let $(\mathcal{G}, P_1, bP_1, sbP_1, P_2, (s+\mu)P_1)$ be the instance of DDH1 that $\mathscr{B}_1$ has to solve i.e., decide whether $\mu = 0$ or $\mu \xleftarrow{\text{U}} \mathbb{Z}_p$. The phases of the game are simulated by $\mathscr{B}_1$ as described below.

Setup: Choose $c, d, u, \Delta_1, \Delta_3, \Delta_4, (e_j, \Delta_{2,j})_{j=1}^{h} \xleftarrow{\text{U}} \mathbb{Z}_p$ and set parameters as:

$U_1 = -\Delta_1(bP_1) + dP_1$, $V_{1,j} = -\Delta_{2,j}(bP_1) + e_jP_1$ for $j = 1, \ldots, h$, $W_1 = -\Delta_3(bP_1) + cP_1$,
$g_T = e(bP_1, P_2)^{-\Delta_4}e(P_1, P_2)^u$
$\mathcal{PP} : (P_1, bP_1, U_1, (V_{1,j})_{j=1}^{h}, W_1, g_T)$

All the secret scalars present in the $\mathcal{MSK}$ are known. $\mathscr{B}_1$ can thus create normal keys. However, $\mathscr{B}_1$'s lack of knowledge of the scalar b or its encoding in $\mathbb{G}_2$ does not allow it to create semi-functional keys.

Key Generation Phases 1 & 2: $\mathscr{B}_1$ answers all of $\mathscr{A}$'s queries with normal keys generated by the $\mathcal{H}_1.\mathsf{KeyGen}$ algorithm.

Challenge: $\mathscr{A}$ sends two message-identity pairs $(m_0, \widehat{\mathbf{id}}_0), (m_1, \widehat{\mathbf{id}}_1)$. $\mathscr{B}_1$ chooses $\beta \xleftarrow{\text{U}} \{0, 1\}$, encrypts M_β under $\widehat{\mathbf{id}}_\beta$ and sends the resulting ciphertext $\widehat{\mathcal{C}} = (\widehat{C}_0, \widehat{C}_1, \widehat{C}_2, \widehat{C}_3, \widehat{\mathsf{tag}})$ to $\mathscr{A}$. Let $\widehat{\mathbf{id}}_\beta = (\widehat{\mathsf{id}}_1, \ldots, \widehat{\mathsf{id}}_{\widehat{\ell}})$. $\widehat{\mathcal{C}}$ is computed as:

$\widehat{\mathsf{tag}} \xleftarrow{\text{U}} \mathbb{Z}_p$,
$\widehat{C}_0 = M_\beta \cdot e(sbP_1, P_2)^{-\Delta_4}e((s+\mu)P_1, P_2)^u = M_\beta \cdot g_T^s e(P_1, P_2)^{u\mu}$,
$\widehat{C}_1 = (s+\mu)P_1 = sP_1 + \mu P_1$,
$\widehat{C}_2 = sbP_1$,
$\widehat{C}_3 = (-\Delta_1 - \sum_{j=1}^{\widehat{\ell}} \Delta_{2,j}\widehat{\mathsf{id}}_j - \widehat{\mathsf{tag}} \cdot \Delta_3)(sbP_1) + (d + \sum_{j=1}^{\widehat{\ell}} e_j\widehat{\mathsf{id}}_j + \widehat{\mathsf{tag}} \cdot c)(s+\mu)P_1$
$= (-\Delta_1 b + d + \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j(-\Delta_{2,j}b + e_j) + \widehat{\mathsf{tag}}(-\Delta_3 b + c))(sP_1) + (d + \sum_{j=1}^{\widehat{\ell}} e_j\widehat{\mathsf{id}}_j + \widehat{\mathsf{tag}} \cdot c)(\mu P_1)$
$= s(U_1 + \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j V_{1,j} + \widehat{\mathsf{tag}}W_1) + \mu(d + \sum_{j=1}^{\widehat{\ell}} e_j\widehat{\mathsf{id}}_j + \widehat{\mathsf{tag}} \cdot c)P_1$.

Observe that $\widehat{\mathcal{C}}$ is normal if $\mu = 0$ and semi-functional when $\mu \xleftarrow{\text{U}} \mathbb{Z}_p$.

Guess: $\mathscr{A}$ outputs its guess β' and halts.

$\mathscr{B}$ returns 1 if $\mathscr{A}$'s guess is correct i.e., $\beta = \beta'$; otherwise $\mathscr{B}_1$ returns 0. The advantage of $\mathscr{B}_1$ in solving the DDH1 instance is given by

$$\begin{aligned}\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}}(\mathscr{B}_1) &= |\Pr[\mathscr{B}_1 \text{ returns } 1|\mu = 0] - \Pr[\mathscr{B}_1 \text{ returns } 1|\mu \xleftarrow{\mathrm{U}} \mathbb{Z}_p]| \\ &= |\Pr[\beta = \beta'|\mu = 0] - \Pr[\beta = \beta'|\mu \xleftarrow{\mathrm{U}} \mathbb{Z}_p]| \\ &= |\Pr[X_{real}] - \Pr[X_{0,1}]|.\end{aligned}$$

Since $\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}}(\mathscr{B}_1) \leq \varepsilon_{\mathrm{DDH1}}$, we have $|\Pr[X_{real}] - \Pr[X_{0,1}]| \leq \varepsilon_{\mathrm{DDH1}}$.

Lemma 2. $|\Pr[X_{k-1,1}] - \Pr[X_{k,0}]| \leq \varepsilon_{\mathrm{DDH2}}$.

Proof. $\mathscr{B}_2$ is given an instance $(\mathcal{G}, P_1, P_2, rP_2, cP_2, (rc+\gamma)P_2)$ of DDH2 and asked to decide whether $\gamma = 0$ or $\gamma \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. It simulates the game as described below.

Setup: Pick scalars $u, \Delta'_1, \Delta'_3, \Delta'_4, d_1, d_2, (e_{j,1}, e_{j,2}, \Delta'_{2,j})_{j=1}^{h} \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and $b \xleftarrow{\mathrm{U}} \mathbb{Z}_p^{\times}$ and (implicitly) set

$$d = d_1 + cd_2, \quad \Delta_1 = \frac{\Delta'_1 + d}{b}, \quad \Delta_3 = \frac{\Delta'_3 + c}{b}, \quad \Delta_4 = \frac{\Delta'_4 + u}{b},$$

$$e_j = e_{j,1} + ce_{j,2}, \quad \Delta_{2,j} = \frac{\Delta'_{2,j} + e_j}{b} \quad \text{for } j = 1, \ldots, h.$$

Parameters are generated as follows.

$U_1 = -\Delta'_1 P_1$, $V_{1,j} = -\Delta'_{2,j} P_1$ for $j = 1, \ldots, h$, $W_1 = -\Delta'_3 P_1$,
$g_T = e(P_1, P_2)^{-\Delta'_4}$
$\mathcal{PP} : (P_1, bP_1, U_1, (V_{1,j})_{j=1}^{h}, W_1, g_T)$

The elements $\Delta_1, \Delta_{2,j}, \Delta_3, d, e_j$ that are part of the $\mathcal{MSK}$ are not available to $\mathscr{B}_2$. Even without these, $\mathscr{B}_2$ can generate keys as explained in the simulation of the key generation phases.

Key Generation Phases 1 & 2: $\mathscr{A}$ queries on identities $\mathbf{id}_1, \mathbf{id}_2, \ldots, \mathbf{id}_q$. $\mathscr{B}$ responds to the i-th query ($i \in [1, q]$) considering three cases.

Case 1: $i > k$

$\mathscr{B}_2$ returns a normal key, $\mathcal{SK}_{\mathbf{id}_i} = (\mathcal{S}_1, \mathcal{S}_2)$ with $\mathcal{S}_1 = ((K_i)_{i\in[1,5]}, (D_{1,j}, E_{1,j})_{j\in[\ell+1,h]})$ and $\mathcal{S}_2 = ((J_i)_{i\in[1,5]}, (D_{2,j}, E_{2,j})_{j\in[\ell+1,h]})$. The master secret is not completely available to $\mathscr{B}_2$ and hence the $\mathcal{H}_1$.KeyGen needs a modification. The $\mathcal{S}_1$-components are computed as shown below.

$r_1, r_2 \xleftarrow{\mathrm{U}} \mathbb{Z}_p$,

$$K_1 = r_1 P_2, \;\; K_2 = r_1(cP_2),$$

$$K_3 = \left(u + r_1\left(d_1 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,1}\right)\right) P_2 + r_1\left(d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}\right)(cP_2) = \left(u + r_1\left(d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j\right)\right) P_2,$$

$$K_4 = -b^{-1} r_1(\Delta'_3 P_2 + cP_2) = -r_1\left(\frac{\Delta'_3 + c}{b}\right) P_2 = -r_1 \Delta_3 P_2,$$

$$K_5 = -b^{-1}\left(\Delta'_4 + u + r_1\left(\Delta'_1 + d_1 + \sum_{j=1}^{\ell} \mathsf{id}_j(\Delta'_{2,j} + e_{j,1})\right)\right) P_2 - b^{-1} r_1\left(d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}\right)(cP_2)$$

$$= b^{-1}\left(-\Delta'_4 - u - r_1\left(\Delta'_1 + d + \sum_{j=1}^{\ell} \mathsf{id}_j(\Delta'_{2,j} + e_j)\right)\right) P_2$$
$$= \left(-\frac{\Delta'_4 + u}{b} - r_1\left(\frac{\Delta'_1 + d}{b} + \sum_{j=1}^{\ell} \mathsf{id}_j\left(\frac{\Delta'_{2,j} + e_j}{b}\right)\right)\right) P_2$$
$$= \left(-\Delta_4 - r_1\left(\Delta_1 + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j}\right)\right) P_2,$$

for $j = \ell + 1, \ldots, h$,
$D_{1,j} = r_1(e_{j,1}P_2 + e_{j,2}(cP_2)) = r_1 e_j P_2$,
$E_{1,j} = -r_1 b^{-1}(\Delta'_{2,j} + e_{j,1})P_2 - r_1 b^{-1} e_{j,2}(cP_2) = -r_1\left(\frac{\Delta'_{2,j} + e_j}{b}\right) P_2 = -r_1 \Delta_{2,j} P_2$.

$\mathcal{S}_2$-components are generated in a similar fashion using a randomiser $r_2 \xleftarrow{\text{U}} \mathbb{Z}_p$ and leaving out the scalars u and Δ'_4. Details are omitted.

Case 2: $i < k$

In this case, $\mathscr{B}_2$ first creates a normal key $\mathcal{SK}_{\mathbf{id}_i}$ and runs $\mathcal{H}_1.\mathsf{SFKeyGen}$ on $\mathcal{SK}_{\mathbf{id}_i}$. This is possible because the only scalar used in $\mathcal{H}_1.\mathsf{SFKeyGen}$ is b which is known to $\mathscr{B}_2$.

Case 3: $i = k$

Let $\mathcal{SK}_{\mathbf{id}_k} = (\mathcal{S}_1, \mathcal{S}_2)$ be the key that $\mathscr{B}_2$ generates for $\mathbf{id}_k$. Elements of $\mathcal{S}_2$ are created normally (as indicated in **Case 1**). In the $\mathcal{S}_1$-portion of $\mathcal{SK}_{\mathbf{id}_k}$, $\mathscr{B}_2$ embeds the DDH2 instance (consisting of $P_2, cP_2, rP_2, (rc+\gamma)P_2$) by generating the components as:

$K_1 = rP_2$, $K_2 = (rc + \gamma)P_2$,

$$K_3 = uP_2 + \left(d_1 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,1}\right)(rP_2) + \left(d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}\right)(rc+\gamma)P_2$$
$$= uP_2 + r\left(d_1 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,1} + c\left(d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}\right)\right) P_2 + \gamma\left(d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}\right) P_2$$
$$= \left(u + r\left(d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j\right)\right) P_2 + \gamma\left(d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}\right) P_2,$$

$$K_4 = -b^{-1}(\Delta'_3 r P_2 + (rc+\gamma)P_2) = -r\left(\frac{\Delta'_3 + c}{b}\right) P_2 - \left(\frac{\gamma}{b}\right) P_2 = -r\Delta_3 P_2 - \left(\frac{\gamma}{b}\right) P_2,$$

$$K_5 = -b^{-1}\left(\Delta'_1 + d_1 + \sum_{j=1}^{\ell} \mathsf{id}_j(\Delta'_{2,j} + e_{j,1})\right)(rP_2) - b^{-1}\left(d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}\right)(rc+\gamma)P_2$$
$$= -b^{-1} r\left(\Delta'_1 + d + \sum_{j=1}^{\ell} \mathsf{id}_j(\Delta'_{2,j} + e_j)\right) P_2 - b^{-1}\gamma\left(d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}\right) P_2$$
$$= -r\left(\frac{\Delta'_1 + d}{b} + \sum_{j=1}^{\ell} \mathsf{id}_j\left(\frac{\Delta'_{2,j} + e_j}{b}\right)\right) P_2 - \left(\frac{\gamma}{b}\right)\left(d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}\right) P_2$$

$$= -r \left(\Delta_1 + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j} \right) P_2 - \left(\frac{\gamma}{b} \right) \left(d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2} \right) P_2,$$

for $j = \ell + 1, \ldots, h$,
$D_{1,j} = e_{j,1}(rP_2) + e_{j,2}(rc + \gamma)P_2 = re_jP_2 + \gamma e_{j,2}P_2$,
$E_{1,j} = -b^{-1}(\Delta'_{2,j} + e_{j,1})rP_2 - b^{-1}e_{j,2}(rc + \gamma)P_2$
$= -r \left(\frac{\Delta'_{2,j} + e_j}{b} \right) P_2 - \left(\frac{\gamma e_{j,2}}{b} \right) P_2$
$= -r\Delta_{2,j}P_2 - \left(\frac{\gamma e_{j,2}}{b} \right) P_2.$

When $\gamma = 0$, $\mathcal{SK}_{\mathbf{id}_k}$ is normal with $r_1 = r$; otherwise, it is partial semi-functional with

$r_1 = r$, $\gamma_1 = \gamma$,
$\pi = d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}$ and
$\pi_j = e_{j,2}$ for $j = \ell + 1, \ldots, h$

set implicitly.

Challenge: $\mathscr{B}_2$ obtains two message-identity pairs $(m_0, \widehat{\mathbf{id}}_0), (m_1, \widehat{\mathbf{id}}_1)$ from $\mathscr{A}$. It then picks $\beta \xleftarrow{\mathrm{U}} \{0,1\}$, $s, \mu \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and generates a semi-functional encryption of M_β under $\widehat{\mathbf{id}}_\beta = (\widehat{\mathsf{id}}_1, \ldots, \widehat{\mathsf{id}}_{\widehat{\ell}})$ given by $\widehat{\mathcal{C}} = (\widehat{C}_0, \widehat{C}_1, \widehat{C}_2, \widehat{C}_3, \widehat{\mathsf{tag}})$ where

$$\widehat{\mathsf{tag}} = -d_2 - \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j e_{j,2},$$
$$\widehat{C}_0 = M_\beta \cdot g_T^s \cdot e(P_1, P_2)^{u\mu},$$
$$\widehat{C}_1 = sP_1 + \mu P_1,$$
$$\widehat{C}_2 = sbP_1,$$
$$\widehat{C}_3 = s \left(U_1 + \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j V_{1,j} + \widehat{\mathsf{tag}} W_1 \right) + \mu \left(d_1 + \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j e_{j,1} \right) P_1$$
$$= s \left(U_1 + \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j V_{1,j} + \widehat{\mathsf{tag}} W_1 \right)$$
$$+ \mu \left((d_1 + cd_2) + \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j (e_{j,1} + ce_{j,2}) + \widehat{\mathsf{tag}} \cdot c \right) P_1 - \mu \left(d_2 c + \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j e_{j,2} c \right) P_1 - \widehat{\mathsf{tag}} \cdot c\mu P_1$$
$$= s \left(U_1 + \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j V_{1,j} + \widehat{\mathsf{tag}} W_1 \right)$$
$$+ \mu \left(d + \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j e_j + \widehat{\mathsf{tag}} \cdot c \right) P_1 + c\mu \left(-d_2 - \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j e_{j,2} - \widehat{\mathsf{tag}} \right) P_1$$
$$= s \left(U_1 + \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j V_{1,j} + \widehat{\mathsf{tag}} W_1 \right) + \mu \left(d + \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j e_j + \widehat{\mathsf{tag}} \cdot c \right) P_1.$$

The last step follows due to the fact that $\widehat{\mathsf{tag}} = -d_2 - \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j e_{j,2}$. Note that $\widehat{\mathcal{C}}$ is properly formed. Also, this is the only way $\mathscr{B}_2$ can generate a semi-functional ciphertext since no encoding of c is available in the group $\mathbb{G}_1$. An implication is that $\mathscr{B}_2$ can only create a nominally semi-functional ciphertext for $\mathbf{id}_k$ since the relation $\mathsf{tag} = -\pi$ will hold, thus providing no information to $\mathscr{B}_2$ about the semi-functionality of $\mathcal{SK}_{\mathbf{id}_k}$.

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

$\mathscr{B}_2$ outputs 1 if $\mathscr{A}$ wins and 0 otherwise. Also, $\mathscr{B}_2$ simulates $\mathsf{G}_{k-1,1}$ if $\gamma = 0$ and $\mathsf{G}_{k,0}$ if $\gamma \xleftarrow{\text{U}} \mathbb{Z}_p$. Therefore, the advantage of $\mathscr{B}_2$ in solving the DDH2 instance is given by

$$\begin{aligned}\mathsf{Adv}_{\mathcal{G}}^{\text{DDH2}}(\mathscr{B}_2) &= |\Pr[\mathscr{B}_2 \text{ returns } 1|\gamma = 0] - \Pr[\mathscr{B}_2 \text{ returns } 1|\gamma \xleftarrow{\text{U}} \mathbb{Z}_p]| \\ &= |\Pr[\beta = \beta'|\mu = 0] - \Pr[\beta = \beta'|\mu \xleftarrow{\text{U}} \mathbb{Z}_p]| \\ &= |\Pr[X_{k-1,1}] - \Pr[X_{k,0}]|.\end{aligned}$$

It now follows that $|\Pr[X_{k-1,1}] - \Pr[X_{k,0}]| \leq \varepsilon_{\text{DDH2}}$ from the fact that $\mathsf{Adv}_{\mathcal{G}}^{\text{DDH2}}(\mathscr{B}) \leq \varepsilon_{\text{DDH2}}$ for all t-time adversaries $\mathscr{B}$. What remains is to show that all the information provided to the adversary have the correct distribution. The scalars $b, u, \Delta'_1, \Delta'_3, \Delta'_4, d_1, d_2, (e_{j,1}, e_{j,2}, \Delta'_{2,j})_{j=1}^h$ chosen by $\mathscr{B}_2$ and r, c, γ from the instance are uniformly and independently distributed. As a consequence the following quantities have the correct distribution.

- r_1, γ_1 for the k-th key
- Δ_4, Δ_3
- $d, (e_j)_{j=1}^h$ and hence $\Delta_1, (\Delta_{2,j})_{j=1}^h$

The same scalars also determine $\pi, (\pi_j)_{j=\ell+1}^h$ for k-th identity and $\widehat{\mathsf{tag}}$ for challenge ciphertext which are required to be uniform and independent quantities. We now argue that this is indeed the case. Let $\mathbf{id}_k = (\mathsf{id}_1, \ldots, \mathsf{id}_h)$ and $\widehat{\mathbf{id}}_\beta = (\widehat{\mathsf{id}}_1, \ldots, \widehat{\mathsf{id}}_h)$ where, for convenience we assume that $\mathsf{id}_{\ell+1} = \cdots = \mathsf{id}_h = \widehat{\mathsf{id}}_{\widehat{\ell}+1} = \cdots \widehat{\mathsf{id}}_h = 0$. Without loss of generality, we consider the case $\ell = 1$ since identity vectors are of length at least 1. The quantities $\pi, (\pi_j)_{j=2}^h, \widehat{\mathsf{tag}}$ are given by the following equation.

$$\begin{pmatrix} \pi \\ \pi_2 \\ \vdots \\ \pi_h \\ \widehat{\mathsf{tag}} \end{pmatrix} = \begin{pmatrix} 1 & \mathsf{id}_1 & \mathsf{id}_2 & \mathsf{id}_3 & \mathsf{id}_4 & \cdots & \mathsf{id}_h \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 \\ -1 & -\widehat{\mathsf{id}}_1 & -\widehat{\mathsf{id}}_2 & -\widehat{\mathsf{id}}_3 & -\widehat{\mathsf{id}}_4 & \cdots & -\widehat{\mathsf{id}}_h \end{pmatrix} \begin{pmatrix} d_2 \\ e_{1,2} \\ e_{2,2} \\ \vdots \\ e_{h-1,2} \\ e_{h,2} \end{pmatrix} \tag{1}$$

Observe that

- the first and last rows in the above matrix are linearly independent since identity components are in $\mathbb{Z}_p^\times$ and $\mathbf{id}_k \neq \widehat{\mathbf{id}}_\beta$. All other rows are linearly independent of these two rows. Hence the matrix has rank $h + 1$.
- $d_2, e_{1,2}, \ldots, e_{h,2}$ are information theoretically hidden from $\mathscr{A}$ and also chosen from uniform and independent distributions over $\mathbb{Z}_p$.

Conditioned on these observations, we conclude that $\pi, (\pi_j)_{j=2}^h, \widehat{\mathsf{tag}}$ are uniformly and independently distributed in $\mathscr{A}$'s view.

Lemma 3. $|\Pr[X_{k,0}] - \Pr[X_{k,1}]| \leq \varepsilon_{\text{DDH2}}$.

The proof is similar to that of Lemma 2. The difference is that $\mathscr{B}_2$ creates a partial semi-functional key for $\mathbf{id}_k$, the k-the identity queried by $\mathscr{A}$, and then embeds the DDH2 instance in $\mathcal{S}_2$-portion of the key. $\mathscr{B}_2$ advantage in solving DDH2 will now depend on whether the $\mathscr{A}$ can determine whether $\mathcal{SK}_{\mathbf{id}_k}$ is partial or fully semi-functional.

Lemma 4. $|\Pr[X_{q,1}] - \Pr[X_{final}]| \leq (2q/p)$.

Proof. In $\mathsf{G}_{q,1}$, all the keys returned to $\mathscr{A}$ are semi-functional and so is the challenge ciphertext. To argue that $\Pr[X_{q,1}] = \Pr[X_{final}]|$, we modify the $\mathcal{H}_1.\mathsf{Setup}$ and $\mathcal{H}_1.\mathsf{SFKeyGen}$ algorithms so that the modification results in G_{final} and the distribution of information provided to the adversary before and after the modification are statistically indistinguishable except with probability $2q/p$.

$\mathcal{H}_1.\mathsf{Setup}$: Pick scalars $\Delta'_1, \Delta'_3, \Delta'_4, u, c, d, (e_j, \Delta'_{2,j})_{j=1}^{h} \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and $b \xleftarrow{\mathrm{U}} \mathbb{Z}_p^{\times}$ and compute parameters as:

$U_1 = -\Delta'_1 P_1$, $V_{1,j} = -\Delta'_{2,j} P_1$ for $j = 1, \ldots, h$, $W_1 = -\Delta'_3 P_1$,
$g_T = e(P_1, P_2)^{-\Delta'_4}$
$\mathcal{PP} : (P_1, bP_1, U_1, (V_{1,j})_{j=1}^{h}, W_1, g_T)$

setting

$$\Delta_1 = \frac{\Delta'_1 + d}{b}, \quad \Delta_3 = \frac{\Delta'_3 + c}{b}, \quad \Delta_4 = \frac{\Delta'_4 + u}{b},$$

$$\Delta_{2,j} = \frac{\Delta'_{2,j} + e_j}{b} \quad \text{for } j = 1, \ldots, h.$$

$\mathcal{H}_1.\mathsf{SFKeyGen}$: Choose $r_1, r_2, \pi', \sigma', (\pi'_j, \sigma'_j)_{j=\ell+1}^{h} \xleftarrow{\mathrm{U}} \mathbb{Z}_p$, $\gamma_1, \gamma_2 \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and compute the individual components as follows.

$K_1 = r_1 P_2$, $K_2 = r_1 c P_2 + \gamma_1 P_2$,

$K_3 = \pi' P_2$,

$K_4 = -r_1 \left(\frac{\Delta'_3 + c}{b}\right) P_2 - \left(\frac{\gamma_1}{b}\right) P_2$,

$K_5 = -\frac{1}{b}\left(\pi' + \Delta'_4 + r_1 \left(\Delta'_1 + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta'_{2,j}\right)\right) P_2$,

$J_1 = r_2 P_2$, $J_2 = r_2(cP_2) + \gamma_2 P_2$,

$J_3 = \sigma' P_2$,

$J_4 = -r_2 \left(\frac{\Delta'_3 + c}{b}\right) P_2 - \left(\frac{\gamma_2}{b}\right) P_2$,

$J_5 = -\frac{1}{b}\left(\sigma' + r_2 \left(\Delta'_1 + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j}\right)\right) P_2$,

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

$D_{1,j} = \pi'_j P_2$,

$E_{1,j} = -\left(\frac{r_1 \Delta'_{2,j} + \pi'_j}{b}\right) P_2$,

$D_{2,j} = \sigma'_j P_2$,

$E_{2,j} = -\left(\frac{r_2 \Delta'_{2,j} + \sigma'_j}{b}\right) P_2$.

The setting of $K_3 = \pi' P_2$ fixes the product $\gamma_1 \pi$ that appears in its semi-functional form i.e., $\left(u + r_1 \left(d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j\right) + \gamma_1 \pi\right) P_2$. The other component where π' is used is K_5 that also fixes $\gamma_1 \pi$ in its semi-functional term. It is necessary to ensure that these two are equal. We show below that K_5 is indeed

well-formed in this sense.

$$\begin{aligned}
K_5 &= -\frac{1}{b}\left(\pi' + \Delta'_4 + r_1\left(\Delta'_1 + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta'_{2,j}\right)\right) P_2 \\
&= -\frac{1}{b}\left(u + r_1\left(d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j\right) + \gamma_1\pi + \Delta'_4 + r_1\left(\Delta'_1 + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta'_{2,j}\right)\right) P_2 \\
&= -\frac{1}{b}\left((\Delta'_4 + u) + r_1\left((\Delta'_1 + d) + \sum_{j=1}^{\ell} \mathsf{id}_j (\Delta'_{2,j} + e_j)\right)\right) P_2 + \gamma_1\pi P_2 \\
&= -\left(\frac{\Delta'_4 + u}{b} + r_1\left(\frac{\Delta'_1 + d}{b} + \sum_{j=1}^{\ell} \mathsf{id}_j \left(\frac{\Delta'_{2,j} + e_j}{b}\right)\right)\right) P_2 + \gamma_1\pi P_2 \\
&= -\left(\Delta_4 + r_1\left(\Delta_1 + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j}\right)\right) P_2 + \gamma_1\pi P_2,
\end{aligned}$$

Similarly, setting $D_{1,j} = \pi'_j P_2$ fixes $\gamma_1\pi_j$ since $D_{1,j}$ has the form $r_1 e_j + \gamma_1\pi_j$. $E_{1,j}$ is computed using π'_j and we justify below that is is properly formed.

$$\begin{aligned}
E_{1,j} &= -\left(\frac{r_1\Delta'_{2,j} + \pi'_j}{b}\right) P_2 \\
&= -\left(\frac{r_1\Delta'_{2,j} + r_1 e_j + \gamma_1\pi_j}{b}\right) P_2 \\
&= -r_1\left(\frac{\Delta'_{2,j} + e_j}{b}\right) P_2 - \frac{\gamma_1\pi_j}{b} P_2 \\
&= -r_1\Delta_{2,j} P_2 - \frac{\gamma_1\pi_j}{b} P_2
\end{aligned}$$

The scalars $\pi', (\pi'_j)_{j=1}^h$ define the products $\gamma_1\pi, (\gamma_1\pi_j)_{j=1}^h$ respectively. Since γ_1 is chosen uniformly from $\mathbb{Z}_p$, $\pi, (\pi_j)_{j=1}^h$ are uniformly and independently distributed in $\mathbb{Z}_p$ except when $\gamma_1 = 0$. Similarly, it is possible to show that $J_5, (E_{2,j})_{j=\ell+1}^h$ are well-formed and $\sigma, (\sigma_j)_{j=1}^h$ have the proper distribution given that $\gamma_2 \neq 0$. Furthermore, all elements of the key are generated independent of $u, d, (e_j)_{j=1}^h$ that determines the independence of the ciphertext from the key. Let us now take a look at the challenge ciphertext:

$\widehat{C}_0 = M_\beta \cdot g_T^s \cdot e(P_1, P_2)^{u\mu}$,
$\widehat{C}_1 = sP_1 + \mu P_1$,
$\widehat{C}_2 = sbP_1$,
$\widehat{C}_3 = -s\left(\Delta'_1 + \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j \Delta'_{2,j} + \widehat{\mathsf{tag}}\Delta'_3\right) P_1 + \mu\left(d + \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j e_j + \widehat{\mathsf{tag}} \cdot c\right) P_1$,

where $\widehat{\mathsf{tag}}, \mu, s \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. Recall that $u, d, (e_j)_{j=1}^h$ are chosen independently and uniformly at random from $\mathbb{Z}_p$. Consequently, components $\widehat{C}_0$ and $\widehat{C}_1$ are randomly distributed in $\mathbb{G}_T$ and $\mathbb{G}_1$ respectively. Also these two components are independent of all other information (including keys and public parameters) provided to $\mathscr{A}$. Therefore the bit β is information theoretically hidden from the adversary implying that the resulting game (obtained by modifying $\mathcal{H}_1.\mathsf{SFKeyGen}$) is G_{final}. The distribution of public parameters remains the unchanged. Let F_i denote the event that $\gamma_1 = 0$ or $\gamma_2 = 0$ for an extract query on $\mathbf{id}_i$ (for $i \in [1, q]$). Clearly $\Pr[\mathsf{F}_i] \leq 2/p$. The keys have the correct distribution unless the event $\mathsf{F} = \cup_{i=1}^q \mathsf{F}_i$ occurs. Thus we have $|\Pr[X_{q,1}] - \Pr[X_{final}]| \leq \Pr[\mathsf{F}] \leq \sum_{i=1}^q \Pr[\mathsf{F}_i] = 2q/p$.

D Security of $\mathcal{H}_2$ - An Overview

The security of $\mathcal{H}_2$ is very similar to that of $\mathcal{H}_1$. We only highlight the main differences and omit the details of the proof.

The definition of semi-functional ciphertexts remains the same. The semi-functional components in keys are defined as for $\mathcal{S}_1$ in $\mathcal{H}_1$. Keys in $\mathcal{H}_2$ do not contain the second set of components $\mathcal{S}_2$. Hence, the notion of partial semi-functionality is not required.

The game sequence is G_{real}, G_0, $(\mathsf{G}_k)_{k=1}^{q}$, G_{final}, where G_{real} is the actual HIBE CPA-security game $\mathsf{ind\text{-}cpa}$ (defined in Section A).In G_0, challenge ciphertext is semi-functional and all keys are normal. G_k, $0 \leq k \leq q$ is similar to G_0 except that the first k keys are semi-functional and the rest are normal. In G_{final}, challenge ciphertext is a semi-functional encryption of a random message and all keys are semi-functional. The theorem below summarises the exact security guarantee obtained for $\mathcal{H}_2$.

Theorem 3. *If* $(\varepsilon_{\text{DDH1}}, t_1)$-DDH1 *and* $(\varepsilon_{\text{DDH2}}, t_2)$-DDH2 *assumptions hold in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *respectively, then* $\mathcal{H}_2$ *is* (ε, t)-$\mathsf{IND\text{-}ID\text{-}CPA}$*-secure where* $\varepsilon \leq \varepsilon_{\text{DDH1}} + q \cdot \varepsilon_{\text{DDH2}} + (q/p)$, $t_1 = t + O(h\rho)$ *and* $t_2 = t + O(h\rho)$. ρ *is the maximum time required for one scalar multiplication in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$.

Since the structure of the ciphertext in $\mathcal{H}_2$ and $\mathcal{H}_1$ are identical, so is the first reduction (based on DDH1). The second reduction is also similar; it is only needed to show that the elements in $\mathbb{G}_2$ that are made public can indeed be generated. The third reduction has one difference. We no longer need to argue about the independence of all information provided to the attacker with respect to the elements $d, (e_j)_{j \in [1,h]}$. In $\mathcal{H}_1$, this was required to show anonymity i.e, the hash of the identity is masked by a random quantity. We only need to show that the message to be masked by a random quantity in the last game and this is done by arguing that the adversary's view (excluding the challenge ciphertext) is independent of the scalar u.