(Anonymous) Compact HIBE From Standard Assumptions

Somindu C. Ramanna Palash Sarkar

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

Abstract

We present 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 $\mathcal{H}_1$ is anonymous and $\mathcal{H}_2$ 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 decision Symmetric eXternal Diffie-Hellman (SXDH) assumption. In terms of provable security properties, all previous constructions of constant-size ciphertext HIBE schemes had one or more of the following drawbacks: secure 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. Along with theoretically satisfying security, the parameter sizes and efficiencies of the different algorithms of the two schemes compare very well with all previously known constructions. Based on currently known techniques, $\mathcal{H}_1$ and $\mathcal{H}_2$ fill an important gap in the state-of-the-art on efficient (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 [Sha84] and the first constructions were proposed by [Coc01, BF03]. In order to reduce the communication and computation overhead of the PKG, [GS02, HL02] introduced Hierarchical IBE (HIBE). 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 schemes are needed? We argue below that all previous 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 in the constructions of HIBE schemes.

Efficiency. A pairing is a bilinear, non-degenerate and efficiently computable map $e : \mathbb{G}_1 \times \mathbb{G}_2$ 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. Practical constructions of HIBE schemes are obtained from such maps.

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 [CM11, SV07, GPS08]. 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 [GS02] and later works [BB04, Wat05, CS06a, Wat09, RCS12], 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 [BBG05]. 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 [CS06b, CS07, LW10, SKOS09, DCIP10, PL13, LPL13, RS13]. Since we are interested in CC-HIBE, we do not consider the line of work [GS02, BB04, Wat05, CS06a, Wat09, RCS12] where the length of the ciphertext depends on the length of the identity tuple.

Security. There are several security-related issues that 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 [BF03, GS02, SW08]. The first HIBE construction [GS02] (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 [BBG05] is secure only under this weaker model.

Anonymity: Another important security notion is *anonymity* [ABC$^+$05] which requires that a ciphertext does not reveal any information about the recipient's identity. Anonymous (H)IBE schemes are useful in constructing public key encryption with keyword search (PEKS) which further extends to more sophisticated primitives such as public key encryption with temporary keyword search (PETKS) and identity-based encryption with keyword search (IBEKS) [ABC$^+$05].

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 decision Diffie-Hellman (DDH) problem over appropriate groups have widespread use in cryptography. Other examples are the decision bilinear Diffie-Hellman (DBDH) problem, the decision linear (DLin) problem and the decision 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, such assumptions are sometimes parametrised by a quantity (such as the maximum length of an identity tuple) arising in the construction. 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 [GH09] all HIBE schemes suffered from a degradation which is exponential in the depth of the HIBE. The construction in [GH09] is very complicated and the security is based on an unnatural assumption. The first practical method of constructing HIBE schemes whose security does not degrade with the depth of the HIBE is due to Waters [Wat09] who introduced the very important technique of dual-system encryption. The work [LW10] provided the first CC-HIBE scheme following the dual-system approach.

The known non-anonymous CC-HIBE schemes are listed in Table 1. It can be seen that all schemes prior to $\mathcal{H}_2$ had one or more of the following security drawbacks: selective-id secure, non-standard/non-static assumption, degradation exponential in the depth of the HIBE. Similarly, the known anonymous HIBE schemes are listed in Table 2 and $\mathcal{H}_1$ is the only scheme among these that achieves all the properties mentioned above. We emphasise that the provable security properties achieved for $\mathcal{H}_1$ and $\mathcal{H}_2$ have not been simultaneously achieved earlier, either for composite-order pairings, or, for prime-order pairings.

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 [JR13] (abbreviated JR-IBE). It is quite natural that the construction of a HIBE scheme will be based on an IBE scheme. Below we mention 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 the simulation, it is essential to ensure that the semi-functional components have proper distributions. The discussion below mentions the known IBE schemes which have security proofs based on the dual-system approach and the difficulties in extending these to CC-HIBE.

The IBE constructions of Waters [Wat09] and its variants [RCS12] 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 [LW10] 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 non-standard assumptions. Two works [LPL13, RS13] independently obtained a CC-HIBE scheme from Lewko-Waters' IBE in prime-order groups. Again, the drawback is that the security of the scheme is based on non-standard assumptions.

Another IBE scheme following the dual-system approach is due to Chen *et.al.* [CLL$^+$12]. This work uses *dual pairing vector spaces* (DPVSs) [OT08, OT09]. 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 [Lew12]. The Chen *et.al.* IBE can be seen as a translation of Lewko-Waters' composite-order pairing-based IBE [LW10] 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 contains vectors from such spaces, the constant-size feature cannot be attained.

More recently, Chen and Wee [CW13] have introduced new techniques for parameter-hiding in DPVS-based constructions. The work describes an IBE scheme and mentions that the full version will describe a compact HIBE. At the time of the writing of this paper, the full version of [CW13] had not appeared and a request to the authors about the details of the HIBE did not receive any response. So, at this point of time, we are not able to determine the possible HIBE scheme mentioned in [CW13] and compare its efficiency and security to that of $\mathcal{H}_1$.

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 [BBG05] (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 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 [LPL13, RS13], anonymity appears as a by-product of the HIBE extension. The basic difficulty 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 which are 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 [BW06] 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.

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, there is no scheme comparable to $\mathcal{H}_2$. The security of the construction in [LW10] is based on sub-group decision assumptions that cannot be considered to be standard assumptions. Table 2 compares $\mathcal{H}_1$ with all previously proposed anonymous HIBE schemes. Again, in terms of security, there is no construction that is comparable to $\mathcal{H}_1$.

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

Scheme	[BBG05]	[CS06b]	[CS07]	[LW10]	$\mathcal{H}_2$
Pairing	Type-1	Type-1	Type-1	Composite	Type-3
Security	selective-id	adaptive-id	selective$^+$-id	adaptive-id	adaptive-id
Assump.	Decision h-wBDHI	h-wDBDHI*	h-wDBDHI*	Subgroup Decision	XDH
Deg.	1	$O((kq2^{N/k})^h)$	1	$O(q)$	$O(q)$
#pp	$(h+4,0)$	$(h+3+hk,0)$	$(2h+3,1)$	$(h+3,1)$	$(3h+9,1)$
#msk	1	1	1	1	2
#cpr	2	2	3	2	3
#key	$h-\ell+2$	$(k+1)(h-\ell)+2$	$2(h-\ell+1)$	$h-\ell+2$	$2(h-\ell)+5$
Enc	$(\ell+2,1)$	(2,1)	$(\ell+2,1)$	$(\ell+2,1)$	$(\ell+4,1)$
Dec	2	2	2	2	3
KGen	$h+2$	$2(h-\ell+1)$	$2h-\ell+2$	$2h-\ell+4$	$2h+5$
Deleg.	$\ell+2$	$2(h-\ell)$	$2h-\ell+1$	$2h-\ell+6$	$2h+9$

Table 1: Comparison of non-anonymous CC-HIBE schemes based on pairings without random oracles. h: maximum depth of the HIBE; ℓ: length of the identity tuple; q: no. of key extraction queries; N ([CS06b]): number of bits in an identity; k ([CS06b]): number of blocks of N/k bits; #pp, #msk, #cpr, #key - number of group elements in the public parameters, master secret, ciphertext and key respectively. Enc, Dec, KGen, Deleg - efficiency of encryption, decryption, key generation and delegation algorithms. 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 of $\mathbb{G}_1$ and b elements of $\mathbb{G}_T$; Enc = (a,b): a scalar multiplications in $\mathbb{G}_1$ and b exponentiations in $\mathbb{G}_T$; Dec: #pairings; KGen: #scalar multiplications in $\mathbb{G}_2$; Deleg: #scalar multiplications in $\mathbb{G}_2$. Assump: underlying complexity assumptions; Deg: security degradation; Zero-ID: whether the scheme allows zero to be an identity component or not.

Scheme	[BW06]	[SKOS09]	[DCIP10]	[PL13]	[LPL13],[RS13]	$\mathcal{H}_1$
Pairing	Type-1	Composite	Composite	Type-1	Type-3	Type-3
Security	selective-id	selective-id	adaptive-id	selective-id	adaptive-id	adaptive-id
Assump.	DLin,DBDH	ℓ-wBDH*, ℓ-cDH	Subgroup Decision	h-BDHE Aug. h-DLin	LW1,LW2,DBDH [LPL13]:3-DH,XDH [RS13]:A1	XDH
Deg.	$O(1)$	$O(1)$	$O(q)$	$O(1)$	$O(q)$	$O(q)$
#pp	$(2(h^2+3h+2),1)$	$(h+6,1)$	$(h+4,1)$	$(h+6,1)$	$(3h+6,1)$	$(h+4,1)$
#msk	h^2+5h+7	$h+4$	2	4	$h+6$	$2h+6$
#cpr	$2h+5$	3	2	4	6	3
#key	$(h+3)(3h-\ell+5)$	$3(h-\ell+3)$	$2(h-\ell+2)$	$3(h-\ell+4)$	$6(h-\ell+2)$	$4(h-\ell)+10$
Enc	$(2(\ell+3)(h+2)+1,1)$	$(\ell+6,1)$	$(\ell+4,1)$	$(\ell+5,1)$	$(3(\ell+2),1)$	$(\ell+4,1)$
Dec	$2h+3$	4	2	4	6	3
KGen	$h^3+h^2(5-\ell)+$ $h(7-3\ell)-2\ell+2$	$3h-2\ell+2$	$4(h+2-3\ell)$	$(h+2(h-\ell+8))$	$6h-5\ell+12$	$2(2h-2\ell+5)$
Deleg.	$5(h+2)(h+3)+1$	$6(h-\ell)+21$	$4(h-\ell)+11$	$(4(h-\ell)+25)$	$2(h-\ell+3)$	$4(h-\ell+5)$

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

of [CHKM10], 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 [LW10] 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.

From the two tables and the above discussion, one can conclude that among anonymous HIBE schemes, $\mathcal{H}_1$ is the most efficient scheme with all the standard provable properties; and that among non-anonymous HIBE schemes a similar statement can be made about $\mathcal{H}_2$. Regarding comparison to non-CC HIBE schemes, clearly $\mathcal{H}_1$ and $\mathcal{H}_2$ will be superior in terms of lower ciphertext size. Further, even though we do not provide the details, we do note that the other parameters of $\mathcal{H}_1, \mathcal{H}_2$ also compare very favourably to important previous non-CC HIBE constructions satisfying similar security [Wat09, RCS12].

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 $x_1, \ldots, x_k \in_{\mathrm{R}} \mathcal{X}$ (or $x_1, \ldots, x_k \xleftarrow{\mathrm{R}} \mathcal{X}$) indicates that elements $x_1, \ldots, x_k$ are sampled independently from the set $\mathcal{X}$ according to some distribution R. The two notation are used interchangeably. U denotes the uniform distribution. 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 written additively and $\mathbb{G}_T$, a multiplicatively written group, all having the same order p and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear, non-degenerate and efficiently computable map.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 decision 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: $P_1 \xleftarrow{\mathrm{U}} \mathbb{G}_1^{\times}$; $P_2 \xleftarrow{\mathrm{U}} \mathbb{G}_2^{\times}$, $a, s \xleftarrow{\mathrm{U}} \mathbb{Z}_p$, $\gamma \xleftarrow{\mathrm{U}} \mathbb{Z}_p$; $\mathcal{D} = (\mathcal{G}, P_1, aP_1, asP_1)$. The advantage of $\mathscr{A}$ in solving the DDH1 problem is given by

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

Essentially, $\mathscr{A}$ has to decide whether $\gamma = 0$ or $\gamma \in_{\mathrm{U}} \mathbb{Z}_p$ given $(\mathcal{D}, (s+\gamma)P_1)$. 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: $P_1 \xleftarrow{\mathrm{U}} \mathbb{G}_1^{\times}$; $P_2 \xleftarrow{\mathrm{U}} \mathbb{G}_2^{\times}$, $r, c \xleftarrow{\mathrm{U}} \mathbb{Z}_p$, $\mu \xleftarrow{\mathrm{U}} \mathbb{Z}_p$;

$$\mathcal{D} = (\mathcal{G}, P_1, P_2, rP_2, cP_2).$$

$\mathscr{A}$'s advantage in solving the DDH2 problem is given by

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

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 [JR13] (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 NIZK via the idea of signatures, the scheme results in ciphertext elements being in $\mathbb{G}_1$.

This section describes a "dual" of the Jutla-Roy [JR13] (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 $P_1 \xleftarrow{\mathrm{U}} \mathbb{G}_1^{\times}$, $P_2 \xleftarrow{\mathrm{U}} \mathbb{G}_2^{\times}$, $\Delta_1, \Delta_2, \Delta_3, \Delta_4, c, d, u, e \xleftarrow{\mathrm{U}} \mathbb{Z}_p$, $b \xleftarrow{\mathrm{U}} \mathbb{Z}_p^{\times}$, 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:

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

Key:

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

Note: In JR-IBE [JR13], 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.

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 $V_{1,1}, \dots, V_{1,h}$ – 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 $\mathbf{id} = (\mathsf{id}_1, \dots, \mathsf{id}_\ell)$, 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 $\mathbf{v}_1 = (d, e_{2,1}, \dots, e_{2,h})$ and $\mathbf{v}_2 = (\Delta_1, \Delta_{2,1}, \dots, \Delta_{2,h})$. 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.

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.

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

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

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

$\mathcal{H}_1.\mathsf{Encrypt}(\mathcal{PP}, m, \mathbf{id} = (\mathsf{id}_1, \dots, \mathsf{id}_\ell))$: 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

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

$\mathcal{H}_1.\mathsf{KeyGen}(\mathcal{MSK}, \mathbf{id} = (\mathsf{id}_1, \dots, \mathsf{id}_\ell))$: 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

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

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

$\mathcal{H}_1.\mathsf{Delegate}(\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell), \mathsf{id}_{\ell+1})$: 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.

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

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

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

setting $r_1 \leftarrow r_1 + \tilde{r}_1 r_2$ and $r_2 \leftarrow \tilde{r}_2 r_2$. 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 $\mathcal{H}_1.\mathsf{KeyGen}$ algorithm.

$\mathcal{H}_1.\mathsf{Decrypt}(\mathcal{C}, \mathcal{SK}_{\mathbf{id}})$: Return m' computed as:

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

Correctness: 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 $\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell)$.

$$\begin{aligned}
&\frac{C_0 \cdot e(C_3, K_1)}{e(C_1, \mathsf{tag} K_2 + K_3) e(C_2, \mathsf{tag} K_4 + K_5)}\\
&= \frac{m \cdot g_T^s \cdot e(s(U_1 + \sum_{j=1}^{\ell} \mathsf{id}_j V_{1,j} + \mathsf{tag} W_1), r_1 P_2)}{e(sP_1, \mathsf{tag} \cdot r_1 c P_2 + (u + r_1(d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j)) P_2) \cdot e(sbP_1, -\mathsf{tag} \cdot r_1 \Delta_3 P_2 - (\Delta_4 + r_1(\Delta_1 + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j})) P_2)}\\
&= \frac{m \cdot g_T^s \cdot e(-\Delta_1 b - \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j} b - \mathsf{tag} \Delta_3 b + d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j + \mathsf{tag} \cdot c, P_2)^{r_1 s}}{e(sP_1, (u + \mathsf{tag} \cdot r_1 c) P_2 + r_1(d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j) P_2) \cdot e(sP_1, -\Delta_4 b - \mathsf{tag} \cdot r_1 \Delta_3 b P_2 - r_1(\Delta_1 b + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j} b) P_2)}\\
&= \frac{m \cdot g_T^s \cdot e(-\Delta_1 b - \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j} b - \mathsf{tag} \Delta_3 b + d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j + \mathsf{tag} \cdot c, P_2)^{r_1 s}}{e(P_1, (u - \Delta_4 b) P_2)^s \cdot e(sP_1, P_2)^{\mathsf{tag} \cdot r_1 c + r_1(d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j)} \cdot e(sP_1, P_2)^{-\mathsf{tag} \cdot r_1 \Delta_3 b - r_1(\Delta_1 b + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j} b)}}\\
&= \frac{m \cdot g_T^s \cdot e((-\Delta_1 b - \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j} b - \mathsf{tag} \Delta_3 b + d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j + \mathsf{tag} \cdot c) P_1, P_2)^{r_1 s}}{g_T^s \cdot e((-\Delta_1 b - \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j} b - \mathsf{tag} \Delta_3 b + d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j + \mathsf{tag} \cdot c) P_1, P_2)^{r_1 s}}\\
&= m.
\end{aligned}$$

The above holds as well for all $\mathcal{SK}_{\mathbf{id}}$ derived from secret keys for higher level identities through the $\mathcal{H}_1.\mathsf{Delegate}$ algorithm.

4.2 Scheme $\mathcal{H}_2$

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.

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

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

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

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

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

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

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

$\mathcal{H}_2.\mathsf{Delegate}(\mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell), \mathsf{id}_{\ell+1})$: 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.

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

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

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.

$\mathcal{H}_2.\mathsf{Decrypt}(\mathcal{C}, \mathcal{SK}_{\mathbf{id}})$: 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:

1. 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 $\mathcal{H}_1$.

2. The KeyGen and $\mathsf{Delegate}$ 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 $\mathscr{H}_2$ 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 $\mathscr{H}_2$ is the only known HIBE within the dual system framework which has rerandomisable keys. The same holds for the corresponding IBE as well.

5 Security of $\mathscr{H}_1$

The scheme $\mathscr{H}_1$ is proved secure in the sense of ANO-IND-ID-CPA (described in Appendix A.3). We first provide algorithms $\mathscr{H}_1$.SFEncrypt and $\mathscr{H}_1$.SFKeyGen that generate semi-functional ciphertexts and keys (respectively) required for the dual system proof. In addition, we need an algorithm PSFKeyGen that generates partial semi-functional keys.

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

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

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

$\mathscr{H}_1.\mathsf{SFKeyGen}(\mathcal{MSK},\mathcal{SK}_{\mathbf{id}})$: 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.

$$\mu_1, \mu_2, \pi, \sigma, (\pi_j, \sigma_j)_{j=1}^{h} \xleftarrow{\text{U}} \mathbb{Z}_p,$$

$$K_1 \leftarrow K_1, \quad K_2 \leftarrow K_2 + \mu_1 P_2, \quad K_3 \leftarrow K_3 + \mu_1 \pi P_2, \quad K_4 \leftarrow K_4 - \left(\frac{\mu_1}{b}\right) P_2, \quad K_5 \leftarrow K_5 - \left(\frac{\mu_1 \pi}{b}\right) P_2,$$
$$D_{1,j} \leftarrow D_{1,j} + \mu_1 \pi_j P_2, \quad E_{1,j} \leftarrow E_{1,j} - \left(\frac{\mu_1 \pi_j}{b}\right) P_2 \quad \text{for } j = \ell + 1, \ldots, h,$$

$$J_1 \leftarrow J_1, \quad J_2 \leftarrow J_2 + \mu_2 P_2, \quad J_3 \leftarrow J_3 + \mu_2 \sigma P_2, \quad J_4 \leftarrow J_4 - \left(\frac{\mu_2}{b}\right) P_2, \quad J_5 \leftarrow J_5 - \left(\frac{\mu_2 \sigma}{b}\right) P_2,$$
$$D_{2,j} \leftarrow D_{2,j} + \mu_2 \sigma_j P_2, \quad E_{2,j} \leftarrow E_{2,j} - \left(\frac{\mu_2 \sigma_j}{b}\right) P_2 \quad \text{for } j = \ell + 1, \ldots, h,$$

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

$\mathsf{PSFKeyGen}(\mathcal{MSK},\mathcal{SK}_{\mathbf{id}})$: Returns a key $\mathcal{SK}_{\mathbf{id}}$ for identity $\mathbf{id}$ with $\mathcal{S}_1$-components having semi-functional terms (generated according to $\mathscr{H}_1$.SFKeyGen algorithm) and $\mathcal{S}_2$-components being normal (as returned by $\mathscr{H}_1$.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 $e(P_1, P_2)^{\mu\gamma(\mathsf{tag}+\pi)}$ 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 $\mathscr{H}_1$.

Theorem 5.1. *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* $\mathscr{H}_1$ *is* (ε, t)-ANO-IND-ID-CPA-*secure where* $\varepsilon \leq \varepsilon_{\text{DDH1}} + 2q \cdot \varepsilon_{\text{DDH2}}$, $t_1 = t + O(h\rho)$ *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. 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 $\mathsf{ano\text{-}ind\text{-}cpa}$ (described in Appendix A.3).
- $\mathsf{G}_{k,0}$, $1 \leq k \leq q$: challenge ciphertext is semi-functional; first $k-1$ keys are semi-functional and k-th key is partial semi-functional.
- $\mathsf{G}_{k,1}$, $0 \leq k \leq q$: 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 5.1, 5.2, 5.3 and 5.4, we show that $|\Pr[X_{real}] - \Pr[X_{0,1}]| \leq \varepsilon_{\text{DDH1}}$, $|\Pr[X_{k-1,1}] - \Pr[X_{k,0}]| \leq \varepsilon_{\text{DDH2}}$, $|\Pr[X_{k,0}]-\Pr[X_{k,1}]| \leq \varepsilon_{\text{DDH2}}$ and $\Pr[X_{q,1}] = \Pr[X_{final}]$ respectively. The advantage of $\mathscr{A}$ in breaking the security of $\mathscr{H}_1$ is thus given by

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

□

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

Proof Sketch: $\mathscr{B}_1$ simulates the game using a DDH1 instance $(\mathcal{G}, P_1, bP_1, sbP_1, P_2, (s+\gamma)P_1)$. The element b of the instance correspond to 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 an encoding of b in $\mathbb{G}_2$ is not 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 (of the instance). $\widehat{\mathcal{C}}$ will be normal or semi-functional depending on whether the instance is real i.e., $\gamma = 0$, or random ($\gamma \in_{\text{U}} \mathbb{Z}_p$). Details of the proof can be found in Appendix B.

Lemma 5.2. $|\Pr[X_{k-1,1}] - \Pr[X_{k,0}]| \le \varepsilon_{\text{DDH2}}$.

Proof Sketch: The DDH2-solver $\mathscr{B}_2$ obtains an instance $(\mathcal{G}, P_1, P_2, rP_2, cP_2, (rc+\mu)P_2$. 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 and 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. The encodings 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 $\mu = 0$, the key will be normal. Otherwise the key is partial semi-functional and μ corresponds to the randomiser μ_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. Refer to Appendix C for details.

Lemma 5.3. $|\Pr[X_{k,0}] - \Pr[X_{k,1}]| \le \varepsilon_{\text{DDH2}}$.

The proof is similar to that of Lemma 5.2. 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 5.4. $|\Pr[X_{q,1}] = \Pr[X_{final}]|$.

Proof Sketch: It is required to show that $\mathsf{G}_{q,1}$ and G_{final} are statistically indistinguishable from the attacker's point of view. 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. The full proof is provided in Appendix D

6 Note on the Security of $\mathscr{H}_2$

The security of $\mathscr{H}_2$ is very similar to that of $\mathscr{H}_1$. 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 Appendix A.2). In G_0, challenge ciphertext is semi-functional and all keys are normal. G_k, $0 \le k \le q$ 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 6.1. *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* $\varepsilon \le \varepsilon_{\text{DDH1}} + q \cdot \varepsilon_{\text{DDH2}}$, $t_1 = t + O(h\rho)$ *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.

7 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 groups 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.

References

[ABC+05] 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.

[BB04] 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.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity-based encryption with constant size ciphertext. In Cramer [Cra05], pages 440–456. Full version available at Cryptology ePrint Archive; Report 2005/015.

[BF03] 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.

[BW06] 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.

[CHKM10] 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.

[CLL$^+$12] 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.

[CM11] Sanjit Chatterjee and Alfred Menezes. On cryptographic protocols employing asymmetric pairings – the role of ψ revisited. *Discrete Applied Mathematics*, 159(13):1311–1322, 2011.

[Coc01] 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.

[Cra05] 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.

[CS06a] 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/.

[CS06b] 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.

[CS07] 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.

[CW13] Jie Chen and Hoeteck Wee. Fully, (almost) tightly secure ibe and dual system groups. In Ran Canetti and Juan A. Garay, editors, *CRYPTO (2)*, volume 8043 of *Lecture Notes in Computer Science*, pages 435–460. Springer, 2013.

[DCIP10] 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.

[Duc10] 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.

[GH09] 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.

[GPS08] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[GS02] 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.

[HL02] 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.

[JR13] Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive nizk proofs for linear subspaces. Cryptology ePrint Archive, Report 2013/109, 2013. `http://eprint.iacr.org/`.

[Lew12] Allison B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 318–335. Springer, 2012.

[LPL13] 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.

[LW10] 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.

[OT08] 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.

[OT09] 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.

[PL13] 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.

[RCS12] 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.

[RS13] Somindu C. Ramanna and Palash Sarkar. Anonymous constant-size ciphertext hibe from asymmetric pairings. Cryptology ePrint Archive, Report 2012/057, 2013. `http://eprint.iacr.org/`, To appear in IMACC 2013.

[Sha84] 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.

[SKOS09] Jae Hong Seo, Tetsutaro Kobayashi, Miyako Ohkubo, and Koutarou Suzuki. Anonymous hierarchical identity-based encryption with constant size ciphertexts. In Stanislaw Jarecki and Gene Tsudik, editors, *Public Key Cryptography*, volume 5443 of *Lecture Notes in Computer Science*, pages 215–234. Springer, 2009.

[SV07] Nigel P. Smart and Frederik Vercauteren. On computable isomorphisms in efficient asymmetric pairing-based systems. *Discrete Applied Mathematics*, 155(4):538–547, 2007.

[SW08] 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.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In Cramer [Cra05], pages 114–127.

[Wat09] 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.

A Hierarchical Identity-Based Encryption

A.1 Definition

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

- Setup: based on an input security parameter κ, generates and outputs the public parameters $\mathcal{PP}$ and the master secret $\mathcal{MSK}$.
- KeyGen: inputs an identity vector **id** and master secret $\mathcal{MSK}$ and outputs the secret key $\mathcal{SK}_{\mathbf{id}}$ corresponding to **id**.
- Encrypt: inputs an identity **id**, a message M and returns a ciphertext $\mathcal{C}$.
- 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 $(\mathsf{id}_1, \ldots, \mathsf{id}_{\ell+1})$.
- Decrypt: inputs a ciphertext $\mathcal{C}$, an identity vector **id**, secret key $\mathcal{SK}_{\mathbf{id}}$ and returns either the corresponding message M or $\perp$ indicating failure.

A.2 CPA-Security

Security against a chosen plaintext attack for HIBE schemes is modelled by the following security game, called ind-cpa [GS02].

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 **id**, the challenger responds with a key $\mathcal{SK}_{\mathbf{id}}$.

Challenge: $\mathscr{A}$ provides two message m_0, m_1 and 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 **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

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

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 $\mathsf{Adv}^{\text{ind-cpa}}_{\text{HIBE}}(\mathscr{A}) \leq \varepsilon$.

A.3 CPA-Security and 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 CPA-security and anonymity and has been used earlier in [Duc10, DCIP10].

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 **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 **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

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

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 $\mathsf{Adv}^{\text{ano-ind-cpa}}_{\text{HIBE}}(\mathscr{A}) \leq \varepsilon$.

B Proof of Lemma 5.1

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

Setup: Choose $c, d, u, \Delta_1, \Delta_3, \Delta_4, (e_j, \Delta_{2,j})_{j=1}^{h} \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ 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 $\mathcal{H}_1$.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:

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

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

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

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

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

C Proof of Lemma 5.2

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

Setup: Pick scalars $u, \Delta_1', \Delta_3', \Delta_4', d_1, d_2, (e_{j,1}, e_{j,2}, \Delta_{2,j}')_{j=1}^{h} \xleftarrow{\text{U}} \mathbb{Z}_p$ and $b \xleftarrow{\text{U}} \mathbb{Z}_p^\times$ 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 $\mathbf{id}_1, \mathbf{id}_2, \dots, \mathbf{id}_q$. $\mathscr{B}$ 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.

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

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

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

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

$$\begin{aligned} K_5 &= -b^{-1}\left(\Delta'_4 + u + r_1\left(\Delta'_1 + d_1 + \sum_{j=1}^{\ell} \mathsf{id}_j(\Delta'_{2,j} + e_{j,1})\right)\right)P_2 - b^{-1} r_1\left(d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}\right)(cP_2) \\ &= b^{-1}\left(-\Delta'_4 - u - r_1\left(\Delta'_1 + d + \sum_{j=1}^{\ell} \mathsf{id}_j(\Delta'_{2,j} + e_j)\right)\right)P_2 \\ &= \left(-\frac{\Delta'_4 + u}{b} - r_1\left(\frac{\Delta'_1 + d}{b} + \sum_{j=1}^{\ell} \mathsf{id}_j\left(\frac{\Delta'_{2,j} + e_j}{b}\right)\right)\right)P_2 \\ &= \left(-\Delta_4 - r_1\left(\Delta_1 + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j}\right)\right)P_2, \end{aligned}$$

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

$D_{1,j} = r_1(e_{j,1}P_2 + e_{j,2}(cP_2)) = r_1 e_j P_2,$

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

$\mathcal{S}_2$-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 $\mathcal{H}_1$.SFKeyGen on $\mathcal{SK}_{\mathbf{id}_i}$. This is possible because the only scalar used in $\mathcal{H}_1$.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 $P_2, cP_2, rP_2, (rc + \mu)P_2$) by generating the components as:

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

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

$$K_4 = -b^{-1}(\Delta'_3 rP_2 + (rc + \mu)P_2) = -r\left(\frac{\Delta'_3 + c}{b}\right)P_2 - \left(\frac{\mu}{b}\right)P_2 = -r\Delta_3 P_2 - \left(\frac{\mu}{b}\right)P_2,$$

$$\begin{aligned}
K_5 &= -b^{-1}\left(\Delta'_1 + d_1 + \sum_{j=1}^{\ell} \mathsf{id}_j(\Delta'_{2,j} + e_{j,1})\right)(rP_2) - b^{-1}\left(d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}\right)(rc + \mu)P_2 \\
&= -b^{-1}r\left(\Delta'_1 + d + \sum_{j=1}^{\ell} \mathsf{id}_j(\Delta'_{2,j} + e_j)\right)P_2 - b^{-1}\mu\left(d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}\right)P_2 \\
&= -r\left(\frac{\Delta'_1 + d}{b} + \sum_{j=1}^{\ell} \mathsf{id}_j\left(\frac{\Delta'_{2,j} + e_j}{b}\right)\right)P_2 - \left(\frac{\mu}{b}\right)\left(d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}\right)P_2 \\
&= -r\left(\Delta_1 + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j}\right)P_2 - \left(\frac{\mu}{b}\right)\left(d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}\right)P_2,
\end{aligned}$$

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

$$\begin{aligned}
D_{1,j} &= e_{j,1}(rP_2) + e_{j,2}(rc + \mu)P_2 = re_j P_2 + \mu e_{j,2} P_2, \\
E_{1,j} &= -b^{-1}(\Delta'_{2,j} + e_{j,1})rP_2 - b^{-1}e_{j,2}(rc + \mu)P_2 \\
&= -r\left(\frac{\Delta'_{2,j} + e_j}{b}\right)P_2 - \left(\frac{\mu e_{j,2}}{b}\right)P_2 \\
&= -r\Delta_{2,j}P_2 - \left(\frac{\mu e_{j,2}}{b}\right)P_2.
\end{aligned}$$

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

$r_1 = r,\ \mu_1 = \mu,$
$\pi = d_2 + \sum_{j=1}^{\ell} \mathsf{id}_j e_{j,2}$ 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 $\beta \xleftarrow{\mathrm{U}} \{0,1\}$, $s, \gamma \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ 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

$$\begin{aligned}
\widehat{\mathsf{tag}} &= -d_2 - \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j e_{j,2}, \\
\widehat{C}_0 &= m_\beta \cdot g_T^s \cdot e(P_1, P_2)^{u\gamma}, \\
\widehat{C}_1 &= sP_1 + \gamma P_1, \\
\widehat{C}_2 &= sbP_1, \\
\widehat{C}_3 &= s\left(U_1 + \textstyle\sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j V_{1,j} + \widehat{\mathsf{tag}} W_1\right) + \gamma\left(d_1 + \textstyle\sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j e_{j,1}\right)P_1
\end{aligned}$$

$$
\begin{aligned}
&= s\left(U_1 + \textstyle\sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j V_{1,j} + \widehat{\mathsf{tag}} W_1\right) \\
&\quad +\gamma\left((d_1 + cd_2) + \textstyle\sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j (e_{j,1} + ce_{j,2}) + \widehat{\mathsf{tag}} \cdot c\right) P_1 - \gamma\left(d_2 c + \textstyle\sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j e_{j,2} c\right) P_1 - \widehat{\mathsf{tag}} \cdot c\gamma P_1 \\
&= s\left(U_1 + \textstyle\sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j V_{1,j} + \widehat{\mathsf{tag}} W_1\right) \\
&\quad +\gamma\left(d + \textstyle\sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j e_j + \widehat{\mathsf{tag}} \cdot c\right) P_1 + c\gamma\left(-d_2 - \textstyle\sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j e_{j,2} - \widehat{\mathsf{tag}}\right) P_1 \\
&= s\left(U_1 + \textstyle\sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j V_{1,j} + \widehat{\mathsf{tag}} W_1\right) + \gamma\left(d + \textstyle\sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j e_j + \widehat{\mathsf{tag}} \cdot c\right) P_1.
\end{aligned}
$$

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 $\mu = 0$ and $\mathsf{G}_{k,0}$ if $\mu \in_{\mathrm{U}} \mathbb{Z}_p$. Therefore, the advantage of $\mathscr{B}_2$ in solving the DDH2 instance is given by

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

It now follows that $|\Pr[X_{k-1,1}] - \Pr[X_{k,0}]| \leq \varepsilon_{\mathrm{DDH2}}$ from the fact that $\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH2}}(\mathscr{B}) \leq \varepsilon_{\mathrm{DDH2}}$ 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, μ from the instance are uniformly and independently distributed. As a consequence the following quantities have the correct distribution.

- r_1, μ_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.

D Proof of Lemma 5.4

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 $\mathscr{H}_1.\mathsf{Setup}$ and $\mathscr{H}_1.\mathsf{SFKeyGen}$ 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.

$\mathscr{H}_1.\mathsf{Setup}$: Pick scalars $\Delta_1', \Delta_3', \Delta_4', u, c, d, (e_j, \Delta_{2,j}')_{j=1}^{h} \xleftarrow{\text{U}} \mathbb{Z}_p$ and $b \xleftarrow{\text{U}} \mathbb{Z}_p^{\times}$ 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

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

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

$\mathscr{H}_1.\mathsf{SFKeyGen}$: Choose $r_1, r_2, \pi', \sigma', (\pi_j', \sigma_j')_{j=\ell+1}^{h} \xleftarrow{\text{U}} \mathbb{Z}_p$, $\mu_1, \mu_2 \xleftarrow{\text{U}} \mathbb{Z}_p^{\times}$ and compute the individual components as follows.

$$K_1 = r_1 P_2, \; K_2 = r_1 c P_2 + \mu_1 P_2, \qquad J_1 = r_2 P_2, \; J_2 = r_2(cP_2) + \mu_2 P_2,$$

$$K_3 = \pi' P_2, \qquad J_3 = \sigma' P_2,$$

$$K_4 = -r_1 \left(\frac{\Delta_3' + c}{b} \right) P_2 - \left(\frac{\mu_1}{b} \right) P_2, \qquad J_4 = -r_2 \left(\frac{\Delta_3' + c}{b} \right) P_2 - \left(\frac{\mu_2}{b} \right) P_2,$$

$$K_5 = -\frac{1}{b} \left(\pi' + \Delta_4' + r_1 \left(\Delta_1' + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j}' \right) \right) P_2, \qquad J_5 = -\frac{1}{b} \left(\sigma' + r_2 \left(\Delta_1' + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j} \right) \right) P_2,$$

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

$$D_{1,j} = \pi_j' P_2, \qquad D_{2,j} = \sigma_j' P_2,$$

$$E_{1,j} = -\left(\frac{r_1 \Delta_{2,j}' + \pi_j'}{b} \right) P_2, \qquad E_{2,j} = -\left(\frac{r_2 \Delta_{2,j}' + \sigma_j'}{b} \right) P_2.$$

The setting of $K_3 = \pi' P_2$ fixes the product $\mu_1 \pi$ that appear in its semi-functional form i.e., $\left(u + r_1 \left(d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j\right) + \mu_1 \pi\right) P_2$. The other component where π' is used is K_5 that also fixes $\mu_1 \pi$ 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.

$$\begin{aligned}
K_5 &= -\frac{1}{b}\left(\pi' + \Delta_4' + r_1\left(\Delta_1' + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j}'\right)\right)P_2 \\
&= -\frac{1}{b}\left(u + r_1\left(d + \sum_{j=1}^{\ell} \mathsf{id}_j e_j\right) + \mu_1\pi + \Delta_4' + r_1\left(\Delta_1' + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j}'\right)\right)P_2 \\
&= -\frac{1}{b}\left((\Delta_4' + u) + r_1\left((\Delta_1' + d) + \sum_{j=1}^{\ell} \mathsf{id}_j(\Delta_{2,j}' + e_j)\right)\right)P_2 + \mu_1\pi P_2 \\
&= -\left(\frac{\Delta_4' + u}{b} + r_1\left(\frac{\Delta_1' + d}{b} + \sum_{j=1}^{\ell} \mathsf{id}_j\left(\frac{\Delta_{2,j}' + e_j}{b}\right)\right)\right)P_2 + \mu_1\pi P_2 \\
&= -\left(\Delta_4 + r_1\left(\Delta_1 + \sum_{j=1}^{\ell} \mathsf{id}_j \Delta_{2,j}\right)\right)P_2 + \mu_1\pi P_2,
\end{aligned}$$

Similarly, setting $D_{1,j} = \pi_j' P_2$ fixes $\mu_1\pi_j$ since $D_{1,j}$ has the form $r_1 e_j + \mu_1\pi_j$. $E_{1,j}$ is computed using π_j' and we justify below that is is properly formed.

$$\begin{aligned}
E_{1,j} &= -\left(\frac{r_1\Delta_{2,j}' + \pi_j'}{b}\right)P_2 \\
&= -\left(\frac{r_1\Delta_{2,j}' + r_1 e_j + \mu_1\pi_j}{b}\right)P_2 \\
&= -r_1\left(\frac{\Delta_{2,j}' + e_j}{b}\right)P_2 - \frac{\mu_1\pi_j}{b}P_2 \\
&= -r_1\Delta_{2,j}P_2 - \frac{\mu_1\pi_j}{b}P_2
\end{aligned}$$

The scalars $\pi', (\pi_j')_{j=1}^{h}$ define the products $\mu_1\pi, (\mu_1\pi_j)_{j=1}^{h}$ respectively. Since μ_1 is chosen uniformly from $\mathbb{Z}_p^{\times}$, $\pi, (\pi_j)_{j=1}^{h}$ are uniformly and independently distributed in $\mathbb{Z}_p$. 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.

So, the scalars $\pi, \sigma, (\pi_j, \sigma_j)_{j=\ell+1}^{h}$ are implicitly set to independent random values in $\mathbb{Z}_p$. Furthermore, all the elements 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:

$\widehat{C}_0 = m_\beta \cdot g_T^s \cdot e(P_1, P_2)^{u\gamma}$,
$\widehat{C}_1 = sP_1 + \gamma P_1$,
$\widehat{C}_2 = sbP_1$,
$\widehat{C}_3 = -s\left(\Delta_1' + \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j \Delta_{2,j}' + \widehat{\mathsf{tag}}\Delta_3'\right)P_1 + \gamma\left(d + \sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j e_j + \widehat{\mathsf{tag}} \cdot c\right)P_1$,

where $\widehat{\mathsf{tag}}, \gamma, s \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. 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 $\mathscr{H}_1$.$\mathsf{SFKeyGen}$) is G_{final}. Also, since the distribution of keys and parameters remains the same, the two games $\mathsf{G}_{q,1}$ and G_{final} are statistically indistinguishable.