Practical hybrid (hierarchical) identity-based encryption schemes based on the decisional bilinear Diffie-Hellman assumption

Sanjit Chatterjee*

Department of Computer Science and Automation,
Indian Institute of Science,
Bangalore, 560012, India
E-mail: sanjit@csa.iisc.ernet.in
*Corresponding author

Palash Sarkar

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

Abstract: The paper proposes a (H)IBE scheme, which improves upon Waters' scheme from Eurocrypt 2005 by significantly reducing the size of the public parameters. The improvement is based on two ideas: 1) partitioning the identities in smaller blocks; 2) reusing public parameters. Modification of the basic HIBE scheme yields a CCA-secure hybrid HIBE where symmetric key authentication is used to eliminate costly pairing operations from the decryption algorithm. The protocols and the security arguments are recast in the most efficient asymmetric pairing setting where one gets several variants of the basic protocol with associated trade-offs in the ciphertext overhead and public parameter size. For practical security levels, the variants we obtain are currently the most efficient and practical among all other schemes which achieve similar security under the DBDH assumption. The basic idea provides improvements to the construction of other cryptographic primitives such as signatures, wildcard IBE and certificateless encryption.

Keywords: identity-based encryption; IBE; adaptive identity attacks; identity-based signature; IBS; chosen ciphertext attacks; asymmetric pairing; decision bilinear Diffie-Hellman problem.

Reference to this paper should be made as follows: Chatterjee, S. and Sarkar, P. (2013) 'Practical hybrid (hierarchical) identity-based encryption schemes based on the decisional bilinear Diffie-Hellman assumption', *Int. J. Applied Cryptography*, Vol. 3, No. 1, pp.47–83.

Biographical notes: Sanjit Chatterjee is an Assistant Professor in the Department of Computer Science and Automation, Indian Institute of Science, Bangalore. His primary research interest is in applied cryptography, in particular, the interplay of theory and practice for provably secure cryptographic protocols.

Palash Sarkar received his Bachelor of Electronics and Telecommunication Engineering degree in 1991 from Jadavpur University, Kolkata and Master of Technology in Computer Science in 1993 from Indian Statistical Institute, Kolkata. He completed his PhD from Indian Statistical Institute in 1999. Since 2005, he has been a Professor at Indian Statistical Institute. His research interests include cryptology, discrete mathematics and computer science.

1 Introduction

The concept of identity-based encryption (IBE) was introduced by Shamir (1984). An IBE is a type of public key encryption where the public key can be any binary string. The corresponding secret key is generated by a private key generator (PKG) and provided to the relevant user. The notion of IBE simplifies several applications of public key cryptography. The first efficient implementation and an appropriate security model for IBE was provided by Boneh and Franklin (2003).

The PKG issues a private key associated with an identity. The notion of hierarchical identity-based encryption (HIBE) was introduced in Horwitz and Lynn (2002) and Gentry and Silverberg (2002) to reduce the workload of the PKG. An entity in a HIBE structure has an identity which is a tuple $(\mathsf{v}_1,\ldots,\mathsf{v}_j)$. The private key corresponding to such an identity can be generated by the entity whose identity is $(\mathsf{v}_1, \ldots, \mathsf{v}_{j-1})$ and which possesses a private key corresponding to this identity. The security model for IBE was extended to that of HIBE in Horwitz and Lynn (2002) and Gentry and Silverberg (2002).

Security of all public key cryptographic protocols relies on some hardness assumption. Most IBE schemes use a bilinear map $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Known examples of such maps arise from elliptic curves where $\mathbb{G}_1$ and $\mathbb{G}_2$ are subgroups of points of an elliptic curve (EC) and $\mathbb{G}_T$ is a subgroup of the multiplicative group of a finite field. In the elliptic curve literature, it is customary to write the group operation additively and we follow this convention for $\mathbb{G}_1$ and $\mathbb{G}_2$. The basic hardness assumption in the setting of bilinear map is the hardness of the bilinear Diffie-Hellman (BDH) problem and the hardness of its decisional version, the so-called decision BDH (DBDH) problem.

The construction of the IBE scheme in Boneh and Franklin (2003) and of the HIBE scheme in Gentry and Silverberg (2002) were proven to be secure in the appropriate models and were based on the hardness of the (D)BDH problem. However, the proofs made use of the so-called *random oracle* heuristic, i.e., the schemes made use of certain functions which are modelled as uniform random functions in the security proof. This led to a search for schemes which can be proven to be secure without the use of random oracles. The first such construction was presented in Canetti et al. (2004). Unfortunately, the work in Canetti et al. (2004) had to relax the security model and consider a weaker model called the selective-ID (sID) model. A more efficient construction of a (H)IBE secure in the sID model was given by Boneh and Boyen (2004a).

The first construction of an IBE which can be proved to be secure without using the random oracle heuristic was given by Boneh and Boyen (2004b). Later, Waters (2005) presented an efficient construction of an IBE which is secure in the same setting.

One disadvantage of Waters' (2005) scheme is the rather large size of the public parameters of the scheme. If identities are represented by a bit string of length n, then the scheme requires a vector of length $(n + 1)$ to be maintained as part of public parameter, where each element of the vector is a point in a suitable elliptic curve group. For κ-bit security, n has to be at least 2κ so that attaining 80-bit (resp. 128-bit) security requires 161 (resp. 257) EC points as part of the public parameters. The rather large size of the public parameters can be a problem for practical implementation of the scheme.

In the same paper, Waters (2005) also outlined a construction of a HIBE. The idea is to have a new set of public parameters for each of the h levels of the HIBE. This leads to a system having $h(n + 1)$ many elliptic curve points as public parameters for an h-level HIBE having n-bit identities at each level.

Bellare and Ristenpart (2009) performed a new analysis of Waters' IBE scheme which does not require the so-called 'artificial abort' step in the original proof of Waters and followed in subsequent work (Chatterjee and Sarkar, 2005; Naccache, 2007). This results in a reduction of the simulation time required in the security proof. On the other hand, the security bound that they obtain is different from the bound obtained by Waters. As a result, the analysis in Bellare and Ristenpart (2009) does not always improve the analysis performed by Waters (2005). Prior to the work of Bellare and Ristenpart (2009), Hofheinz and Kiltz (2008) proposed a new proof technique for Waters IBE. However, as noted in Bellare and Ristenpart (2009), their technique does not provide any significant gain in terms of concrete security.

1.1 Our contributions

The basic problem that we consider in this paper is to construct a practical (H)IBE of small depth for the following setting.

1.1.1 Setting

1. security in the full model (i.e., against adaptive ciphertext and adaptive identity attacks) introduced in Boneh and Franklin (2003)
2. security is based on the hardness of the DBDH problem
3. the proof does not use the random oracle heuristic.

We present solutions to this problem which build upon Waters' (2005) original proposal. Some of the basic ideas behind these solutions appear in our previous conference papers (Chatterjee and Sarkar, 2005, 2006a; Sarkar and Chatterjee, 2007).

The (H)IBE construction is in two parts – construction of a (H)IBE which is secure against chosen plaintext attacks (CPA-secure) and then its modification to ensure security against chosen ciphertext attacks (CCA-secure). The CPA-secure HIBE is then converted to a hierarchical identity-based signature (HIBS) scheme. Below we provide an overview of our main results.

Use of asymmetric pairing. The basic scheme due to Waters as well as most of the follow-up[2] works (Chatterjee and Sarkar, 2005, 2006a; Naccache, 2007; Kiltz and Galindo, 2009) use symmetric pairings. In terms of efficient implementation, this, however, is not a good choice. The recent work by Bellare and Ristenpart (2009) describe Waters (2005) IBE scheme using asymmetric pairings but, in the so-called Type 2 setting. This requires an efficiently-computable isomorphism ψ from $\mathbb{G}_2$ to $\mathbb{G}_1$. They make use of this isomorphism in the security proof. Research on pairing implementation shows (Galbraith et al., 2008) that it is more efficient to implement pairing-based schemes using asymmetric pairings in the so-called Type 3 setting. These are asymmetric pairings for which the isomorphism ψ mentioned above is not known. As a result, it is required to describe the scheme *and* the security proof for the scheme without using such a mapping. The Type 3 setting turns out to be the most general setting for pairing-based cryptography.

In this work, we describe all constructions using Type 3 pairings. This naturally leads to several variants with associated trade-offs. We carefully analyse these variants and identify the advantages of each. The use of Type 3 pairings is not completely routine and requires quite a bit of thought to identify the useful ideas. Since the (H)IBE scheme and associated primitives originating from the work of Waters' form an important class of cryptographic primitives, we believe that our use of Type 3 pairings is of significant value to the practical implementation of the relevant primitives.

CPA-secure (H)IBE construction.

1 For an n-bit identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_n)$, Waters (2005) defines a hash of the form $U_1' + \sum_{i=1}^{n} \mathsf{v}_i U_i$, where $U_1', U_1, \ldots, U_n$ are EC points and are part of the public parameters of the scheme. The generalisation proposed in our conference paper (Chatterjee and Sarkar, 2005) is to consider v as consisting of l blocks of n/l-bits, $v_1, \ldots, v_l$ where each block is considered to be an integer in the range $\{0, \ldots, 2^{n/l} - 1\}$. The corresponding hash is then $U_1' + \sum_{i=1}^{l} v_i U_i$. The number of EC points required in this case is $l + 1$. Putting $l = n$ gives Waters scheme, whereas choosing a suitable l gives a scheme requiring lesser space for the public parameters. There is a consequent negative effect on the security which we discuss later.

2 The extension to HIBE as suggested by Waters (2005), was to choose the EC points $U_1', U_1, \ldots, U_n$ separately for each level of the HIBE. In other words, for an h-level HIBE, the public parameters are of the form $U_1', U_{1,1}, \ldots, U_{1,n}, U_2', U_{2,1}, \ldots, U_{2,n}, \ldots, U_h', U_{h,1}, \ldots, U_{h,n}$. In contrast, the HIBE construction suggested by us in Chatterjee and Sarkar (2006a) uses public parameters of the form $U_1', \ldots, U_h', U_1, \ldots, U_l$ for $1 \leq l \leq n$. In other words, the parameters $U_1', \ldots, U_h'$ correspond to the different levels of the HIBE, whereas the parameters $U_1, \ldots, U_l$ are the same for all the levels. For $l = n$, this suggestion reduces the number of EC points from $(n + 1)h$ to $n + h$ without any additional security degradation over Waters' suggestion.

Signature schemes. It is an observation of Naor, that a CPA-secure IBE scheme can be converted to a signature scheme which is existentially unforgeable under chosen message attacks. This conversion has been used earlier to construct several signature schemes (Boneh et al., 2001; Boneh and Boyen, 2004; Waters, 2005). Our conference paper (Chatterjee and Sarkar, 2005) also shows a conversion of the IBE scheme to a signature scheme. Here we present a more general conversion of the HIBE scheme to obtain a hierarchical identity-based signature (HIBS) scheme. The special case of IBS obtained from the HIBS has smaller size public parameters compared to the earlier (Paterson and Schuldt, 2006) IBS obtained from 2-level Waters (2005) HIBE.

CCA-secure (H)IBE construction. The above mentioned (H)IBE schemes are proved to be secure against adversaries which are restricted from making ciphertext queries. Our conference paper (Sarkar and Chatterjee, 2007) modifies a technique from Boneh et al. (2005b) to ensure security against chosen ciphertext attacks. In this paper, we extend the technique of Sarkar and Chatterjee (2007) to work in the setting of asymmetric pairing. While the basic idea of the conversion is based on Boneh et al. (2005b), some new ideas are introduced:

1 Ciphertext validity checking is partially done through symmetric key authentication techniques. Doing this avoids several pairing computations that would otherwise be required for ciphertext validity checks.

2 Due to the use of symmetric key authentication, the scheme requires hybrid encryption. We propose the first application of efficient single-pass symmetric key authenticated encryption (AE) schemes (Rogaway, 2004; Sarkar, 2010) in the context of hybrid (H)IBE schemes.

Analysis of the schemes. The basic idea of the analysis of CPA-security originates in the work of Boneh and Boyen (2004a, 2004b) where they show a particularly elegant algebraic method to simulate responses to the adversarial key generation queries and to generate the challenge ciphertext. Their construction requires mapping an identity into an EC point. The proposed method for performing this in Boneh and Boyen (2004b) is not very efficient. Waters (2005) proposed an efficient method for doing this but, as mentioned above, still requires large public parameter size [it has been pointed out in Bellare and Ristenpart (2009) that the mapping used by Waters (2005) has actually appeared earlier (Chaum et al., 1987) in a different context]. The analysis in Waters (2005) faced a certain complication due to the possible non-independence of the adversary's success and the simulator's requirement of aborting. This was overcome by using the so-called 'artificial abort' technique which led to an increase in the runtime of the simulator. This technique was later followed in several other papers (Chatterjee and Sarkar, 2005; Naccache, 2007). As mentioned earlier, Bellare and Ristenpart (2009) performed a different analysis of Waters' scheme which does not require the artificial abort step.

We present the detailed analysis of the schemes proposed here following both Waters (2005) and Bellare and Ristenpart (2009). Waters required an aritifical abort step to complete the analysis. This required the simulator to possibly abort even after the adversary has successfully completed its task. Bellare and Ristenpart remove this artificial abort. One would have expected the resulting analysis to have provided a better bound than that obtained by Waters. However, in general the bound in Bellare and Ristenpart (2009) is worse than the bound in Waters (2005). This becomes much more evident when one moves from IBE to HIBE. We nail down the exact reason for this.

Due to the new analysis and due to the use of Type 3 pairings, the proofs in this paper are different from the proofs provided in our conference papers (Chatterjee and Sarkar, 2005, 2006a). Also, Bellare and Ristenpart (2009) work in the Type 2 setting where for the proof they require an efficiently-computable isomorphism ψ from $\mathbb{G}_2$ to $\mathbb{G}_1$. Since we work in the Type 3 setting, we do not require any such isomorphism and so, our proofs are also different from those in Bellare and Ristenpart (2009).

The security argument for the CCA-secure scheme is new to this paper and is not present in our previous conference paper (Sarkar and Chatterjee, 2007). This proof requires to handle several subtleties arising from the fact that the ciphertext wellformedness is checked using symmetric key techniques. In a sense, this is similar to Kurosawa and Desmedt's (2004) hybrid PKE which improves upon the decryption algorithm of Cramer and Shoup's (2003) PKE scheme.

1.2 Related work

The Waters' (2005) IBE scheme is based on the algebraic framework developed by Boneh and Boyen (2004a, 2004b). This framework has later been called the commutative blinding framework (Boyen, 2008a). In the first paper, Boneh and Boyen (2004a) consider the weaker selective-identity security model and develop the proof technique to simulate the response to adversarial key generation queries and to simulate the generation of the challenge ciphertext. The second paper employs this technique to construct a scheme which is secure under adaptive key generation requests from the adversary. The essence of both techniques is to use an appropriate function to hash an identity into an EC point and then use this point to blind the master secret key while generating decryption keys. In a theoretical sense the problem of constructing IBE schemes secure under adaptive attacks was already settled in Boneh and Boyen (2004b). The disadvantage was that the scheme was very inefficient which was essentially due to the form of the function that was used to map identities into an EC point.

The main contribution of Waters (2005) was to design an efficient hash function so as to work with the above mentioned algebraic framework. Since this brought the resulting scheme into the domain of practicability (except for the large size of the public parameters), the work (Waters, 2005) became quite important in stimulating further research works on associated primitives. As mentioned earlier, it has been pointed out in Bellare and Ristenpart (2009) that the hash function used by Waters (2005) was previously used in a different context (Chaum et al., 1987). We have already mentioned our development of Waters' scheme.

Independent of our work, Naccache (2007) has obtained the same generalisation of Waters' IBE scheme as described in our conference paper (Chatterjee and Sarkar, 2005). His work is also in the setting of symmetric pairings. The use of Type 3 pairings, the HIBE schemes, the HIBS signature schemes and the modifications to attain CCA-security are not present in Naccache's work.

In recent years, several important constructions of (H)IBE schemes have appeared in the literature. These are mentioned below and we justify why from a practical point of view, the constructions and the analysis in the current paper are of interest. We would like to point out that all the schemes mentioned below are in the setting of symmetric pairings. The only exceptions are the recent construction of Waters (2009) where the possibility of using asymmetric pairings is mentioned only in the passing and the work of Kiltz and Vahlis (2008) where asymmetric setting is briefly considered towards the end of the paper.

1. Gentry (2006) provided a simple and efficient IBE which can be proved to be secure in the full model without using the random oracle heuristic. But, the assumption used in Gentry (2006) is the so-called q-ABDHE, which is a significantly more complex assumption in comparison to the much simpler DBDH assumption used in this work. One aspect of this complexity is the fact that the assumption depends on q which is the number of private key queries made by an adversary. In the literature, such assumptions are called non-static assumptions.
2. Waters (2009) (note that this is different from Waters, 2005) introduces a new technique called dual-system encryption and presents an IBE where the number of public parameters is constant and does not depend on the size of the security parameter κ. For values of κ of practical interest (such as $\kappa = 80$ or 128), we show that it is possible to obtain public parameter sizes which are smaller or comparable to Waters (2009) with an additional security degradation of only a few bits. Efficiencies of the encryption, decryption and key generation algorithms of the schemes proposed in this work are significantly better than the corresponding algorithms in Waters (2009) (see Table 2 later). Thus, from a practical point of view, the IBE scheme described here is overall more efficient than the asymptotically better scheme in Waters (2009). The same remark applies to other IBE constructions in the dual-system paradigm (Lewko and Waters, 2011; Ramanna et al., 2012; Lewko, 2012).
3. Kiltz and Galindo (2009) have described CCA-secure HIBE schemes by modifying the basic (Waters, 2005) HIBE. Compared to Kiltz and Galindo (2009), our technique for attaining CCA-security leads to a more efficient decryption algorithm. This is due to the use of symmetric key techniques to perform ciphertext validity checking.
4. In a work subsequent to the appearance of our conference paper (Sarkar and Chatterjee, 2007), Kiltz and Vahlis (2008) describe a CCA-secure IBE scheme which uses symmetric key authentication technique (similar to the techniques used in Sarkar and Chatterjee, 2007) and is based on the hardness of the

mBDDH assumption, which is a stronger assumption than the DBDH assumption. They obtain better encryption and decryption efficiency at a cost of increase in the size of the private key and the key generation time.

5 The HIBE scheme described here has a security degradation which is exponential in the length of the HIBE, a feature which is shared by several other constructions. As a consequence, the scheme is suitable only for small depths (around 2 or 3). Some later constructions avoid the exponential security degradation but have other trade-offs. We mention these below.

   a The first such HIBE scheme is given in Gentry and Halevi (2009). This construction, however, is quite complicated and uses a hardness assumption which is significantly more complicated than the assumption q-ABDHE used in Gentry (2006).

   b Waters (2009), extends his IBE scheme to propose a HIBE scheme. While, the construction and the underlying assumptions are simpler than those in Gentry and Halevi (2009), compared to the schemes described here, they are still quite inefficient.

   c Building on the work in Waters (2009), Lewko and Waters (2010) describe a HIBE scheme which achieve constant size ciphertexts. Previously HIBE schemes with this property (Boneh et al., 2005a) had been obtained only for a weaker security model. The downside in Lewko and Waters (2010) is that the scheme is described using pairings over composite order groups. They require the group order to be the product of three distinct primes and the assumption that it is difficult to find any factor of the group order. Since this is based on the difficulty of factoring, the group order must be significantly larger leading to a corresponding loss in efficiency.

6 Starting with the work of Gentry et al. (2008), several IBE schemes have been proposed which use lattices as the underlying algebraic framework. Rapid development of lattice based IBE schemes have to some extent mirrored the development of pairing-based IBE schemes. The basic hardness assumption is that of the learing with errors (LWE) problem. The one major drawback of lattice-based cryptography in general and which is also true for IBE schemes is that the sizes of public parameters and keys are rather large. So, from a practical point of view, lattice based IBE schemes are still not comparable with pairing-based schemes.

To summarise, consider the practical issue of designing a CCA-secure IBE scheme achieving 80-bit or 128-bit security in the setting described above (based on the DBDH assumption and without using the random oracle heuristic). For this problem, the construction (and the analysis) presented here provides the most efficient and practical solution known till date. In the same setting, this paper also describes the most efficient CCA-secure small-depth HIBE schemes for practical security levels. The (identity-based and hierarchical identity-based) signature schemes described here are the most efficient currently known schemes based on the co-CDH problem in the standard model. On the other hand, the constructions presented here might not be particularly suitable for HIBEs of higher depth. Another obvious limitation is that the number of group elements in the public parameter is (sub)linear in the bit-length of the idenity elements (hence, the security parameter). However, as demonstrated in the paper, the latter is not a significant problem for practical constructions.

2 Preliminaries

The running time of all algorithms in this paper is upper bounded by a polynomial in a security parameter κ. Formally, all algorithms take 1^κ as input. We will be assuming this formalism without explicitly mentioning it.

2.1 HIBE scheme

Following Horwitz and Lynn (2002) and Gentry and Silverberg (2002), a HIBE scheme is specified by four (probabilistic) algorithms (which are polynomial time in the security parameter): Set-Up, KeyGen, Encrypt and Decrypt. For a HIBE of height h (henceforth denoted as h-HIBE) any identity v is a tuple $(\mathsf{v}_1, \dots, \mathsf{v}_j)$ where $1 \le j \le h$.

- $\mathsf{HIBE.SetUp}$. Takes as input the security parameter and outputs (pk, sk), where pk is the public parameter of the PKG and sk is the master secret of the PKG. It also defines the domains of identities, messages and ciphertexts.
- $\mathsf{HIBE.KeyGen}(\mathsf{v}, d_{\mathsf{v}|_{j-1}}, pk)$. Takes as input a j-level identity v, the secret $d_{\mathsf{v}|_{j-1}}$ corresponding to its $(j-1)$-level prefix and pk and returns as output d_v, the secret key corresponding to v. In case $j = 1$, $d_{\mathsf{v}|_{j-1}}$ is equal to sk, the master secret of the PKG.
- $\mathsf{HIBE.Encrypt}(\mathsf{v}, M, pk)$. Takes as input v, the message M and pk, and returns C, the ciphertext obtained by encrypting M under v and pk.
- $\mathsf{HIBE.Decrypt}(\mathsf{v}, d_\mathsf{v}, C, pk)$. Takes as input v, the secret key d_v corresponding to v, a ciphertext C and pk. Returns either $\perp$ or M, the message which is the decryption of C.

As usual, for soundness, we require that $\mathsf{HIBE.Decrypt}(\mathsf{v}, d_\mathsf{v}, C, pk) = M$ must hold for all v, d_v, C, pk, sk and M associated by the above four algorithms.

Note. In this paper, a *valid* ciphertext is one which can be produced as an output of the encryption algorithm. A ciphertext is *invalid* if it is not valid. The definition of the decryption algorithm, in itself, does not include a definition of invalid ciphertext and so, the decryption algorithm may never reject any ciphertext. As an example, the decryption algorithm in Kiltz and Galindo (2009) is of this type. Our CCA-secure schemes, however, will reject any invalid ciphertext.

Identity-based encryption. An IBE is a special case of a HIBE where the number of levels h is equal to one.

2.2 Security model for HIBE

Security is defined using an adversarial game. An adversary $\mathcal{A}$ is allowed to query two oracles – a decryption oracle and a key-extraction oracle. The game has several stages.
Set-up. In this stage, the simulator sets up the scheme and provides $\mathcal{A}$ with the public parameters.

Query Phase 1. Adversary $\mathcal{A}$ makes a finite number of queries in an adaptive fashion, where each query is addressed either to the decryption oracle or to the key-extraction oracle. In a query to the decryption oracle, it provides a ciphertext as well as the identity under which it wants the decryption. It gets back the corresponding message or bad if the ciphertext is invalid. Similarly, in a query to the key-extraction oracle, it asks for the private key of the identity it provides and gets back this private key. Further, $\mathcal{A}$ is allowed to make these queries adaptively, i.e., any query may depend on the previous queries as well as their answers. The adversary is not allowed to make any useless queries, i.e., queries for which it can compute the answer itself. For example, the adversary is not allowed to ask for the decryption of a message under an identity if it has already obtained a private key corresponding to the identity.

Challenge. At this stage, $\mathcal{A}$ outputs an identity $\mathsf{v}^* = (\mathsf{v}_1^*, \ldots, \mathsf{v}_j^*)$ for $1 \leq j \leq h$, and a pair of messages M_0 and M_1. There is the natural restriction on the adversary, that it cannot query the key extraction oracle on v^* or any of its proper prefixes in either of the query phases 1 or 2. A random bit γ is chosen and the adversary is provided with C^* which is an encryption of M_γ under v^*.

Query Phase 2. $\mathcal{A}$ now issues additional queries just like Phase 1, with the (obvious) restrictions that it cannot ask the decryption oracle for the decryption of C^* under v^*.

Guess. $\mathcal{A}$ outputs a guess γ' of γ.

The advantage of the adversary $\mathcal{A}$ is defined as: $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HIBE}} = |\Pr[(\gamma = \gamma')] - 1/2|$.

The quantity $\mathsf{Adv}^{\mathsf{HIBE}}(t, q_{\mathsf{ID}}, q_{\mathsf{C}})$ denotes the maximum of $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HIBE}}$ where the maximum is taken over all adversaries running in time at most t and making at most q_{C} queries to the decryption oracle and at most q_{ID} queries to the key-extraction oracle. A HIBE scheme is said to be $(\epsilon, t, q_{\mathsf{ID}}, q_{\mathsf{C}})$-CCA secure if $\mathsf{Adv}^{\mathsf{HIBE}}(t, q_{\mathsf{ID}}, q_{\mathsf{C}}) \leq \epsilon$.

In the above game, we can disallow the adversary $\mathcal{A}$ from querying the decryption oracle. Then $\mathsf{Adv}^{\mathsf{HIBE}}(t, q)$ denotes the maximum advantage where the maximum is taken over all adversaries running in time at most t and making at most q queries to the key-extraction oracle. A HIBE scheme is said to be (t, q, ϵ)-CPA secure if $\mathsf{Adv}^{\mathsf{HIBE}}(t, q) \leq \epsilon$.

Shi and Waters (2008) consider the situation where the distribution of the keys depend on the actual delegation path. We have not included this in the above security model since in our constructions the decryption keys are always uniformly distributed.

Security model for IBE. The security model for IBE is derived from the security model for HIBE by simply allowing only one level in the hierarchy.

2.3 Cryptographic bilinear map

Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be cyclic groups having the same prime order p where we write $\mathbb{G}_1, \mathbb{G}_2$ additively and $\mathbb{G}_T$ multiplicatively. A mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is called a cryptographic bilinear map if it satisfies the following properties.

- *Bilinearity*. $e(aQ, bR) = e(Q, R)^{ab} = e(bQ, aR)$ for all $Q \in \mathbb{G}_1$, $R \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p$.
- *Non-degeneracy*. If $e(Q, R)$ is the identity of $\mathbb{G}_T$, then either Q is the identity of $\mathbb{G}_1$ or R is the identity of $\mathbb{G}_2$.
- *Computability*. There exists an efficient algorithm to compute $e(Q, R)$ for all $Q \in \mathbb{G}_1$ and $R \in \mathbb{G}_2$.

Known examples of bilinear maps have $\mathbb{G}_1$ and $\mathbb{G}_2$ to be subgroups of points on an elliptic curve and $\mathbb{G}_T$ to be a subgroup of the multiplicative group of a finite field. Hence, in papers on implementation of bilinear pairings, it is customary to write $\mathbb{G}_1$ and $\mathbb{G}_2$ additively and $\mathbb{G}_T$ multiplicatively. We follow this convention as it is closer to the known examples.

If $\mathbb{G}_2 = \mathbb{G}_1$, the pairing is called symmetric since it satisfies the property $e(aP, bP) = e(P, P)^{ab} = e(bP, aP)$. Following Galbraith et al. (2008) we call this the Type 1 setting. Most of the pairing-based protocols are usually described in this setting because of its relative simplicity. For example, the initial proposal due to Waters (2005) or its latter variants (Chatterjee and Sarkar, 2005, 2006a) are given in terms of symmetric pairings. It is known, however, that to achieve the same security level the Type 1 setting requires working over larger size fields than those required for the asymmetric settings. For practical implementations this leads to a significant increase in the sizes of the various quantities such as the public parameters and the ciphertexts. Further, this difference in sizes cascades into a difference in the performance of the associated algorithms including encryption and decryption. Also the Type 1 pairings are quite restricted in terms of the choice of curves. Hence, it is deemed desireable that the practical pairing-based schemes should be implemented using asymmetric pairing whenever feasible.

In the asymmetric setting, if an efficiently-computable isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ is known then it is called the Type 2 setting (Galbraith et al., 2008). If no such isomorphism is known then the pairing setting is called the Type 3 setting. Current research (Galbraith et al., 2008) suggests that Type 3 is overall a better choice than Type 2 in terms of size of elements of $\mathbb{G}_2$, the cost of scalar multiplication in $\mathbb{G}_2$, the cost of pairing computation, etc. In this work we use Type 3 setting to describe the protocols and their security arguments. As a result, the schemes can be implemented using fast pairings such as the ate pairing (Hess et al., 2006), the R-ate pairing (Lee et al., 2009) or the optimal ate pairing (Vercauteren, 2010).

The asymmetric setting allows the possibility of having a smaller representation of the elements of $\mathbb{G}_1$ than that of $\mathbb{G}_2$. The relative size depends on the choice of the curve [see Freeman et al. (2010) for the available options of pairing friendly curves]. For an IBE (or signature) scheme, the relative size of the elements of $\mathbb{G}_1$ and $\mathbb{G}_2$ leads to several trade-off questions: which of the four quantities (public parameters, master secret key, decryption key, ciphertext) consist of elements of $\mathbb{G}_1$? Clearly, the best situation would be if we could make all of these to be elements of $\mathbb{G}_1$, but that is not possible. The different possibilities of the trade-off give rise to different variants of the basic scheme as we will see later.

2.4 Hardness assumptions

Let $\mathbb{G}_1 = \langle P_1 \rangle$, $\mathbb{G}_2 = \langle P_2 \rangle$, $\mathbb{G}_T = \langle e(P_1, P_2) \rangle$. Note that in the Type 3 setting no efficiently-computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ is known. We define the decisional bilinear Diffie-Hellman (DBDH) problem in $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$. Given a tuple $(aP_1, aP_2, bP_1, cP_1, cP_2, Z)$, where a, b, c are independent and uniform random elements of $\mathbb{Z}_p$; and $Z \in \mathbb{G}_T$; distinguish between the distribution where $Z = e(P_1, P_2)^{abc}$ (which we denote as Z is real) and the distribution where Z is an independent and uniform random element of $\mathbb{G}_T$ (which we denote as Z is random). The advantage of a probabilistic algorithm $\mathcal{B}$, which takes as input a tuple $(aP_1, aP_2, bP_1, cP_1, cP_2, Z)$ and outputs a bit, in solving the DBDH problem is defined as

$$\begin{aligned} \mathsf{Adv}_{\mathcal{B}}^{\mathrm{DBDH}} = & |\Pr[\mathcal{B}(aP_1, aP_2, bP_1, cP_1, cP_2, Z) \\ & = 1 | \mathsf{Z \text{ is real}}] \\ & -\Pr[\mathcal{B}(aP_1, aP_2, bP_1, cP_1, cP_2, Z) \\ & = 1 | \mathsf{Z \text{ is random}}]| \end{aligned} \quad (1)$$

where the probability is calculated over the random choices of $a, b, c \in \mathbb{Z}_p$ as well as the random bits used by $\mathcal{B}$. The quantity $\mathsf{Adv}^{\mathrm{DBDH}}(t)$ denotes the maximum of $\mathsf{Adv}_{\mathcal{B}}^{\mathrm{DBDH}}$ where the maximum is taken over all adversaries $\mathcal{B}$ running in time at most t. By the (ϵ, t)-DBDH assumption we mean $\mathsf{Adv}^{\mathrm{DBDH}}(t) \leq \epsilon$.

Several versions of the (D)BDH problem in the asymmetric setting are available in the literature (Smart and Vercauteren, 2007; Boyen, 2008b; Chatterjee and Menezes, 2011). The version we use here is termed as DBDH-3 in Chatterjee and Menezes (2011) where it has been shown that DBDH-3 is equivalent to DBDH-2, which is the corresponding problem defined in the Type 2 settings. We also note that the version of the DBDH in Type 2 used in the analysis of Waters protocol in Bellare and Ristenpart (2009) also provides bP_2 as part of the problem instance. This element is dropped from the problem instance of DBDH-3 (as defined above) making it a potentially weaker assumption.

A weaker variant of the DBDH assumption is the following: given $(aP_2, bP_1, bP_2, cP_1, Z)$ determine whether $Z = e(P_1, P_2)^{abc}$ or Z is random. Note that the quantities cP_2 and aP_1 are not provided as part of the input instance. This is a sort of minimalist assumption in the setting of asymmetric pairing and has been called DBDH-3b in Chatterjee and Menezes (2011). Later, we point out that for some of the schemes that we describe, it is sufficient to assume the hardness of the DBDH-3b problem.

The co-CDH problem in $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ is as follows: given a random $Q \in \mathbb{G}_1$ and $(zP_1, zP_2) \in \mathbb{G}_1 \times \mathbb{G}_2$ for a uniform random $z \in \mathbb{Z}_p$, the requirement is to compute zQ [this version of the co-CDH problem in the Type 3 setting has been shown to be equivalent to the co-CDH problem in the Type 2 setting (Chatterjee et al., 2010)]. An algorithm is said to be successful if it can find zQ. The advantage of an algorithm in solving the co-CDH problem is defined to be the probability of its success. The co-CDH problem is said to be (t, ϵ)-hard if the maximum advantage of any adversary running in time t in solving the co-CDH problem is at most ϵ. In the setting of symmetric pairings, i.e., if $\mathbb{G}_1 = \mathbb{G}_2$, then the co-CDH problem is exactly the CDH problem in $\mathbb{G}_1$.

2.5 Components AE, KDF and UOWHF

We briefly introduce and state the security notions for AE, KDF and UOWHF. These will be required to construct the CCA-secure HIBE scheme.

An AE scheme consists of two deterministic algorithms – $\mathsf{Encrypt}$ and $\mathsf{Decrypt}$. Both of these use a common secret key k. The $\mathsf{Encrypt}_k$ algorithm takes as input a nonce IV and a message M and returns (C, tag), where C is (usually) of the same length as M. The $\mathsf{Decrypt}_k$ algorithm takes as input a nonce IV and a pair (C, tag) and returns either the message M or $\perp$ (indicating invalid ciphertext).

An AE algorithm possesses two security properties – privacy and authenticity. For privacy, the adversarial game is the following. The adversary $\mathcal{A}$ is given access to an oracle which is either the encryption oracle instantiated with a random key k or is an oracle which simply returns random strings of length equal to its input. After interacting with the oracle the adversary ultimately outputs a bit. The advantage of $\mathcal{A}$ is defined to be $|\mathsf{Pr}[\mathcal{A}=1|\text{real oracle}]-\mathsf{Pr}[\mathcal{A}=1|\text{random oracle}]|$. In the above game, the adversary is assumed to be nonce-respecting, i.e., it does not repeat a nonce.

The security notion defined above provides the privacy of an AE scheme. In particular, it implies the following notion of one-time security. The adversary submits two equal length messages M_0 and M_1. A random (IV^*, k^*) pair is chosen and a random bit γ is chosen. The adversary is given (C^*, tag^*) which is the encryption of M_γ using IV^* and k^*. The adversary then outputs γ' and its advantage is $|\mathsf{Pr}[\gamma=\gamma']-1/2|$. We say that an AE scheme satisfies (ϵ, t) one-time encryption security if the maximum advantage of any adversary running in time t in the above game is ϵ.

The authenticity property of an AE scheme is defined through the following game. A nonce respecting adversary $\mathcal{A}$ is given access to an encryption oracle instantiated by a secret key k. It submits messages to the oracle and receives as output ciphertext-tag pairs. Finally, it outputs a 'new' ciphertext-tag pair and a nonce, which can be equal to a previous nonce. The advantage of $\mathcal{A}$ in this game is the probability that the forgery is valid, i.e., it will be accepted as a valid ciphertext. By an (ϵ, t)-secure authentication of an AE scheme we mean that the maximum advantage of any adversary running in time t in the above game is ϵ.

A KDF is a function $\mathsf{KDF}()$ which takes an input K and produces as output a string Y. The security notion for KDF is the following. For a randomly chosen K, the adversary has to distinguish between $\mathsf{KDF}(K)$ and a uniform random Y. The advantage of an adversary attacking the KDF property is defined as $|\mathsf{Pr}[\mathcal{A}(\mathsf{KDF}(K))=1]-\mathsf{Pr}[\mathcal{A}(Y)=1]|$. By an (ϵ, t)-KDF we mean that the advantage of any adversary running in time at most t is upper bounded by ϵ.

A function family $\{H_k\}_{k\in\mathcal{K}}$ is said to be a universal one-way hash family if the following adversarial task is difficult. The adversary outputs an x; is then given a randomly chosen $k\in\mathcal{K}$ and has to find $x'\neq x$ such that $H_k(x)=H_k(x')$. We say that the family is (ϵ, t)-UOWHF if the maximum advantage (probability) of an adversary running in time t and winning the above game is ϵ.

2.6 *Notation to denote time and space requirements*

We mention the notation that will be used to denote the times of different operations. Different notation is employed to separate the use of symmetric pairing setting from that of asymmetric pairing setting.

2.6.1 *Asymmetric pairing setting*

Recall that in this case $e:\mathbb{G}_1\times\mathbb{G}_2\to\mathbb{G}_T$.

Time requirements.

- $[\mathrm{SM}_1]$: cost of one scalar multiplication in $\mathbb{G}_1$
- $[\mathrm{SM}_2]$: cost of one scalar multiplication in $\mathbb{G}_2$
- $[\mathrm{P}_j]$: cost of computing a product of j pairings of the form $e(R_1,Q_1)\times e(R_2,Q_2)\times\cdots\times e(R_j,Q_j)$
- $[\mathrm{E}]$: cost of one exponentiation in $\mathbb{G}_T$
- $\left[\mathrm{H}_{n,l}^{(1)}\right]$: cost of hashing an identity into an element of $\mathbb{G}_1$ [see equation (2)]
- $\left[\mathrm{H}_{n,l}^{(2)}\right]$: cost of hashing an identity into an element of $\mathbb{G}_2$ [see equation (3)].

Note that $e(Q_1,R_1)/e(Q_2,R_2)=e(Q_1,R_1)\times e(Q_2,-R_2)$ whose cost is $[\mathrm{P}_2]$. Similarly the cost of ratio of product of pairings of more terms will be denoted by an appropriate $[\mathrm{P}_j]$. The rationale behind this notation is that it may be more efficient to compute the product of pairings together than to separately compute the pairings and then multiply them.

The cost of computing $V_k^{(1)}(v)$ (denoted by $\left[\mathrm{H}_{n,l}^{(1)}\right]$) is given by l scalar multiplications in $\mathbb{G}_1$, but, the size of each scalar is small (only n/l bits as compared to the $\lg p$ bits for a full scalar multiplication). So, one may consider $1\left[\mathrm{H}_{n,l}^{(1)}\right]$ to be approximately equal to $1[\mathrm{SM}_1]$. We, however, separately account for this cost. Similar considerations hold for $\left[\mathrm{H}_{n,l}^{(2)}\right]$.

Space requirements. The sizes of the public parameters; master secret key; a decryption key corresponding to an identity; and a ciphertext will be denoted by a tuple (i_1,i_2,i_3,i_4) which denotes that i_1 elements of $\mathbb{G}_1$, i_2 elements of $\mathbb{G}_2$, i_3 elements of $\mathbb{G}_T$ and i_4 elements of $\mathbb{Z}_p$ are required.

2.6.2 *Symmetric pairing setting*

Recall that in this case $e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$.

Time requirements.

- $[\mathrm{SM}]$: cost of one scalar multiplication in $\mathbb{G}$
- $[\mathrm{SP}_j]$: cost of computing a product of j pairings of the form $e(R_1,Q_1)\times e(R_2,Q_2)\times\cdots\times e(R_j,Q_j)$
- $[\mathrm{SE}]$: cost of one exponentiation in $\mathbb{G}_T$
- $[\mathrm{H}_{n,l}]$: cost of hashing an identity into an element of $\mathbb{G}$ [see equation (2) with $\mathbb{G}=\mathbb{G}_1$)].

Space requirements. The sizes of the public parameters; master secret key; a decryption key corresponding to an identity; and a ciphertext will be denoted by a tuple $((i_1,i_2,i_3))$ which denotes that i_1 elements of $\mathbb{G}$, i_2 elements of $\mathbb{G}_T$ and i_3 elements of $\mathbb{Z}_p$ are required.

3 CPA-secure HIBE construction

In the asymmetric pairing setting, it is possible to obtain different versions of the same protocol that was originally proposed in the symmetric setting. Here we provide two variants of the HIBE scheme in the Type 3 setting each of which allows different optimisation strategies. The first one may be considered to be the basic HIBE scheme whereas the second one is a variant which is advantageous in certain situations, particularly when restricted to IBE.

3.1 HIBE-1

The identities are of the type $(\mathsf{v}_1, \ldots, \mathsf{v}_j)$, for $j \in \{1, \ldots, h\}$ where each $\mathsf{v}_k = (\mathsf{v}_{k,1}, \ldots, \mathsf{v}_{k,l})$ and $\mathsf{v}_{k,i}$ is an (n/l)-bit string which will also be considered to be an integer in the set $\{0, \ldots, 2^{n/l} - 1\}$, where l divides n. Choosing $l = n$ gives v_k to be an n-bit string as originally considered by Waters (2005).

Set-up. The scheme is built from a Type-3 bilinear setting $(p, \mathbb{G}_1 = \langle R_1 \rangle, \mathbb{G}_2 = \langle P_2 \rangle, \mathbb{G}_T, e)$ as mentioned in Section 2.3.

The public parameters are the following elements: P_2, $e(R_1, Q_2)$, $U'_1, \ldots, U'_h$, $U_1, \ldots, U_l$, where $Q_2 = \alpha P_2$ for some α chosen uniformly at random from $\mathbb{Z}_p$ and U'_i and U_js are chosen independently and uniformly at random from $\mathbb{G}_1$. The master secret is $\alpha R_1 \in \mathbb{G}_1$.

Hashing an identity into an element of $\mathbb{G}_1$. Let $v = (v_1, \ldots, v_l)$, where each v_i is an (n/l)-bit string and is considered to be an element of $\mathbb{Z}_{2^{n/l}}$. For $1 \leq k \leq h$ we define,

$$V_k^{(1)}(v) = U'_k + \sum_{i=1}^{l} v_i U_i. \tag{2}$$

The (1) in the superscript refers to the fact that the hashing is done into an element of $\mathbb{G}_1$. The quantities $U'_k, U_1, \ldots, U_l$ are elements of $\mathbb{G}_1$ and are implicit in the notation $V_k^{(1)}(v)$. When v and $\mathbb{G}_1$ are clear from the context we will write V_k instead of $V_k^{(1)}(v)$. The modularity introduced by this notation allows an easier understanding of the scheme. For $l = n$, this form of hashing was used by Waters (2005) (and had appeared earlier in Chaum et al., 1987).

Note that for the j^{th} level of the HIBE, we add a single element, i.e., U'_j in the public parameter while the elements $U_1, \ldots, U_l$ are re-used for each level. This way we are able to shorten the public parameter size.

Key generation. The first level private key is always generated by the PKG using the master secret. For example, the private key corresponding to v_1 will be $d_0 = \alpha R_1 + r_1 V_1^{(1)} \in \mathbb{G}_1$ and $d_1 = r_1 P_2 \in \mathbb{G}_2$ where r_1 is chosen randomly from $\mathbb{Z}_p^*$ and $V_1 = U'_1 + \sum_{j=1}^{l} \mathsf{v}_{1,j} U_j$.

Key delegation can be done in the manner shown in Boneh and Boyen (2004a). Let $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$, $j \leq h$, be the identity for which the private key is required. Suppose $(d'_0, d'_1, \ldots, d'_{j-1})$ is a private key for the identity $(\mathsf{v}_1, \ldots, \mathsf{v}_{j-1})$. To generate a private key for v, choose r_j uniformly at random from $\mathbb{Z}_p$ and compute:

$$d_0 = d'_0 + r_j V_j^{(1)}(\mathsf{v}_j);$$

$$d_i = d'_i \text{ for } 1 \leq i \leq j-1; \quad d_j = r_j P_2.$$

Then $d_{\mathsf{v}} = (d_0, d_1, \ldots, d_j)$ is a private key for v. This means that we can write

$$d_0 = \alpha R_1 + \sum_{k=1}^{j} r_k V_k^{(1)}(\mathsf{v}_k) \text{ and } d_k = r_k P_2 \text{ for } 1 \leq k \leq j$$

where $r_1, \ldots, r_j$ are uniform random elements of $\mathbb{Z}_p$. Note that the private key corresponding to the j^{th} level identity consists of one element from $\mathbb{G}_1$ and j elements from $\mathbb{G}_2$.

Encryption. Let $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$ be the identity under which a message $M \in \mathbb{G}_T$ is to be encrypted. Return

$$\big(C_0 = M \times e(R_1, Q_2)^t, C_1 = tP_2,$$
$$B_1 = tV_1^{(1)}(\mathsf{v}_1), \ldots, B_j = tV_j^{(1)}(\mathsf{v}_j)\big)$$

where t is a uniform random element of $\mathbb{Z}_p$.

Decryption. Let $C = (C_0, C_1, B_1, \ldots, B_j)$ be a ciphertext and $(d_0, d_1, \ldots, d_j)$ be a decryption key for an identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$. Return

$$C_0 \times \left(\prod_{k=1}^{j} \frac{e(B_k, d_k)}{e(d_0, C_1)} \right).$$

If the ciphertext is proper, then the correctness of the decryption follows from a simple calculation using the bilinear property of e.

$$\begin{aligned}
& C_0 \times \frac{\prod_{k=1}^{j} e(B_k, d_k)}{e(d_0, C_1)} \\
&= M \times e(R_1, Q_2)^t \times \frac{\prod_{k=1}^{j} e(B_k, d_k)}{e(d_0, C_1)} \\
&= M \times e(R_1, Q_2)^t \times \frac{\prod_{k=1}^{j} e\left(tV_k^{(1)}(\mathsf{v}_k), r_k P_2\right)}{e\left(\alpha R_1 + \sum_{k=1}^{j} r_k V_k^{(1)}(\mathsf{v}_k), tP_2\right)} \\
&= M \times e(R_1, Q_2)^t \times \frac{\prod_{k=1}^{j} e\left(tV_k^{(1)}(\mathsf{v}_k), r_k P_2\right)}{e\left(\alpha R_1 + \sum_{k=1}^{j} r_k V_k^{(1)}(\mathsf{v}_k), tP_2\right)} \\
&= M \times e(R_1, Q_2)^t \\
&\quad \times \frac{\prod_{k=1}^{j} e\left(tV_k^{(1)}(\mathsf{v}_k), r_k P_2\right)}{e(R_1, Q_2)^t \times e\left(\sum_{k=1}^{j} r_k V_k^{(1)}(\mathsf{v}_k), tP_2\right)} \\
&= M \times e(R_1, Q_2)^t \\
&\quad \times \frac{\prod_{k=1}^{j} e\left(tV_k^{(1)}(\mathsf{v}_k), r_k P_2\right)}{e(R_1, Q_2)^t \times \prod_{k=1}^{j} e(tV_k^{(1)}(\mathsf{v}_k), r_k P_2)} \\
&= M.
\end{aligned}$$

Note. If $h = 1$, then we obtain an IBE scheme and further, if $n = l$, then we obtain Waters' IBE scheme (in the setting of Type 3 pairing).

3.2 HIBE-2

Set-up. The structure of the identities and the setting of Type-3 pairing is the same as that for HIBE-1. The definition of the public parameters and the master secret key is different. Let $\mathbb{G}_1 = \langle P_1 \rangle$ and $\mathbb{G}_2 = \langle P_2 \rangle$. Choose a random element α from $\mathbb{Z}_p$ and define $R_1 = \alpha P_1$. Choose random elements $x_1, \ldots, x_h$ and $y_1, \ldots, y_l$ from $\mathbb{Z}_p$. Choose a random element Q_2 from $\mathbb{G}_2$. The public parameters consist of the following elements.

$$\begin{aligned}
&P_1, U_1' = x_1 P_1, \ldots, U_h' = x_h P_1,\\
&\qquad U_1 = y_1 P_1, \ldots, U_l = y_l P_1,\\
&Q_2, W_1' = x_1 P_2, \ldots, W_h' = x_h P_2,\\
&\qquad W_1 = y_1 P_2, \ldots, W_l = y_l P_2\\
&e(R_1, Q_2) \in \mathbb{G}_T.
\end{aligned}$$

The first and the second rows consist of $h + l + 1$ elements of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. The master secret key is $\alpha Q_2 \in \mathbb{G}_2$.

In the description below, the elements (U_i', U_j) will be required for encryption while the elements (W_i', W_j) will be required for key delegation. The private key will consists of elements of $\mathbb{G}_2$ while the ciphertext will consist of elements of $\mathbb{G}_1$ and $\mathbb{G}_T$.

There are now two kinds of hashes for an n-bit string v which is divided into l strings each of which is n/l bits long. The first kind $V_k^{(1)}$ is defined from $U_k', U_1, \ldots, U_l$ as in equation (2). The other hash is defined as follows. The superscript (2) refers to the fact that hashing is done into $\mathbb{G}_2$ and $W_k', W_1, \ldots, W_l$ are implicit in the definition.

$$V_k^{(2)}(v) = W_k' + \sum_{i=1}^{l} v_i W_i. \tag{3}$$

Key generation. Let $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$, $j \le h$ be the identity for which the private key is required. Key generation and delegation is done as in the HIBE-1. Only difference is that the hash $V_k^{(2)}$ of the component v_k is used instead of the $V_k^{(1)}$. The private key corresponding to v is a tuple $(d_0, d_1, \ldots, d_j)$, where $d_0 = \alpha Q_2 + \sum_{k=1}^{j} r_k V_k^{(2)}(\mathsf{v}_k)$ and $d_k = r_k P_2$.

Encryption. Suppose that a message $M \in \mathbb{G}_T$ is to be encrypted under an identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$. A random element t of $\mathbb{Z}_p$ is chosen and the ciphertext is defined to be

$$\Big(C_0 = M \times e(R_1, Q_2)^t,\\ C_1 = tP_1, B_1 = tV_1^{(1)}, \ldots, B_j = tV_j^{(1)}\Big).$$

Note that other than C_0 all other components are elements of $\mathbb{G}_1$. This allows a reduction in the ciphertext overhead in comparison to HIBE-1 where C_1 is an element of $\mathbb{G}_2$.

Decryption. Given a ciphertext $(C_0, C_1, B_1, \ldots, B_j)$ encrypted under an identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$ and a decryption key d_{v} for v, decryption is performed as follows.

$$C_0 \times \frac{\prod_{k=1}^{j} e(B_k, d_k)}{e(C_1, d_0)}.$$

The correctness of the decryption is seen as follows. First note that from the definition of the U's and the W's the discrete log of $V_k^{(1)}$ to base P_1 is equal to the discrete log of $V_k^{(2)}$ to base P_2. So we can write $V_k^{(1)} = w_k P_1$ and $V_k^{(2)} = w_k P_2$.

$$\begin{aligned}
&C_0 \times \frac{\prod_{k=1}^{j} e(B_k, d_k)}{e(C_1, d_0)}\\
&= M \times e(R_1, Q_2)^t \times \frac{\prod_{k=1}^{j} e(B_k, d_k)}{e\left(tP_1, \alpha Q_2 + \sum_{j=1}^{k} r_k V_k^{(2)}\right)}\\
&= M \times e(R_1, Q_2)^t\\
&\qquad \times \frac{\prod_{k=1}^{j} e(B_k, d_k)}{e(R_1, Q_2)^t \times \prod_{j=1}^{k} e\left(tP_1, r_k V_k^{(2)}\right)}\\
&= M \times \frac{\prod_{k=1}^{j} e(B_k, d_k)}{\prod_{j=1}^{k} e\left(tP_1, r_k w_k P_2\right)}\\
&= M \times \frac{\prod_{k=1}^{j} e(B_k, d_k)}{\prod_{j=1}^{k} e\left(t w_k P_1, r_k P_2\right)}\\
&= M \times \frac{\prod_{k=1}^{j} e(B_k, d_k)}{\prod_{j=1}^{k} e\left(tV_k^{(1)}, d_k\right)}\\
&= M.
\end{aligned}$$

Note. The case $h = 1$ corresponds to an IBE and is of particular interest as a special optimisation is possible in this case. Recall that the $W_i', W_j \in \mathbb{G}_2$ are used in the protocol for the purpose of key delegation only. In an IBE, only the PKG generates the private key and there are no lower level entities which need to have this capability. In this case, the PKG does not need to actually generate the Ws. Instead, the PKG keeps the quantities $x_1, y_1, \ldots, y_l$ as part of the master secret key. The component d_0 of the secret key in the case of IBE equals $\alpha Q_2 + r_1 V_1^{(2)}$. Using the Ws this computation has cost 1[SM$_2$]+1$\left[\mathrm{H}_{n,l}^{(2)}\right]$. If instead the values $x_1, y_1, \ldots, y_l$ are used, then the quantity $r_1 V_1^{(2)}$ can be directly computed as $r_1(x_1 + \sum_{i=1}^{l} \mathsf{v}_i y_i) P_2$ and so the entire computation has cost 1[SM$_2$]. This saves the computation of one $\left[\mathrm{H}_{n,l}^{(2)}\right]$.[3] In other words, in the case of IBE it is possible to reduce the ciphertext overhead from an element of $\mathbb{G}_2$ to an element of $\mathbb{G}_1$ without any associated increase in the public parameter size.

3.3 Security statement and discussion

In this section, we state the result on security and discuss its implications. Two theorems are stated whose proofs are given in Section 4. The first theorem follows Waters (2005) style analysis while the second theorem follows Bellare and Ristenpart (2009) analysis. Note, however, that both Waters (2005) and Bellare and Ristenpart (2009) analyse IBE, whereas, the results below are for HIBE.

Theorem 1. The HIBE schemes described in Section 3.1 and Section 3.2 are (ϵ_{hibe}, t, q)-CPA secure assuming that the (ϵ_{dbdh}, t')-DBDH assumption holds in $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$, with

$$\epsilon_{hibe} \leq 2(m(\mu_l + 1))^h \epsilon_{dbdh};$$

$t' = t + t_{sim}$ where t_{sim} is the simulation time which consists of the time to generate q private keys and one challenge ciphertext plus a time of $O(\epsilon_{hibe}^{-2} \ln(\epsilon_{hibe}^{-1})\lambda^{-1}\ln(\lambda^{-1}))$; $\lambda = 1/(2(m(\mu_l + 1))^h)$; $\mu_l = l(2^{n/l} - 1)$ and $m = \max(2q, 2^{n/l})$. We further assume $m(1 + \mu_l) < p$.

Theorem 2. The HIBE schemes described in Section 3.1 and Section 3.2 are (ϵ_{hibe}, t, q)-CPA secure assuming that the (ϵ_{dbdh}, t')-DBDH assumption holds in $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$, with

$$\epsilon_{hibe} \leq 2(m(\mu_l + 1))^h \epsilon_{dbdh};$$

$t' \leq t + t_{sim}$ and t_{sim} is the simulation time, i.e., the time to generate q private keys and one ciphertext; $\mu_l = l(2^{n/l} - 1)$, and $m = \max(2q/\epsilon_{hibe}, 2^{n/l})$. We further assume $m(1 + \mu_l) < p$.

For $h = 1$, the above theorems state the results for the IBE scheme and we write the corresponding advantage as ϵ_{ibe} (instead of ϵ_{hibe}).

The assumption $m(1 + \mu_l) < p$ in the above statements is practical and similar assumptions are also made in Waters (2005); Naccache (2007), though not quite so explicitly. There are two main differences in the two statements. The first difference is in the nature of the bound on ϵ_{hibe} and this arises from the different expressions for m in the two statements. The second difference is in the nature of the simulation time t_{sim}. The bound is better in the first result while the simulation time is smaller in the second result.

Let $\delta = \delta(n, l, h, q) \stackrel{\Delta}{=} 2(m(\mu_l + 1))^h$. The main point of the above theorems is the bound on ϵ_{hibe}. This is stated to be upper bounded by $\delta\epsilon_{dbdh}$. Ideally, one would like the upper bound to be just ϵ_{dbdh}, since this tightly bounds the adversary's advantage in attacking the HIBE scheme by the advantage in solving the DBDH problem. Viewed in this way, the factor δ indicates the degradation in the bound on adversary's advantage that is promised by the two theorems. Below we simplify the expression for the degradation factor δ.

Since $l \geq 1$, we have $1 + \mu_l = 1 + l(2^{n/l} - 1) \leq l2^{n/l}$. Consequently,

$$\delta(n, l, h, q) = 2(m(\mu_l + 1))^h \leq 2(ml2^{n/l})^h.$$

The degradation is exponential in h and hence the result is meaningful only for small values of h such as 2 or 3. Degradation is determined by the actual value of m which is different for the two theorems.

In Theorem 1, $m = \max(2q, 2^{n/l})$. Typically, one would choose the parameters n and l such that $m = 2q$ and then the bound obtained from Theorem 1 is as follows.

$$\epsilon_{hibe} \leq 2(2ql2^{n/l})^h \epsilon_{dbdh}. \quad (4)$$

In Theorem 2, $m = \max(2q/\epsilon_{hibe}, 2^{n/l})$. For all practical choices of n and l we will have $m = 2q/\epsilon_{hibe}$.

$$\epsilon_{hibe} \leq 2(m(\mu_l + 1))^h \epsilon_{dbdh}$$
$$= 2\left(\frac{2q}{\epsilon_{hibe}}(\mu_l + 1)\right)^h \epsilon_{dbdh} \leq 2\left(\frac{2q}{\epsilon_{hibe}} l2^{n/l}\right)^h \epsilon_{dbdh}.$$

This can be re-written as

$$\epsilon_{hibe}^{h+1} \leq 2(2ql2^{n/l})^h \epsilon_{dbdh}. \quad (5)$$

The quantity q denotes the number of key extraction queries made by the adversary. In the real world, this represents the number of entities who collude to attack the system. A value of q around 2^{30} is more than adequate protection against possible collusion sizes and has been earlier used in Galindo (2004). Here we take q to be 2^{31} so that $2q = 2^{32}$. With this value of q, we suggest that the parameters n and l are chosen so that $2^{n/l} \leq 2^{32}$, so that m equals $2q$. Then the expression for δ becomes $2(4lq2^{n/l})^h$. Putting $h = 1$ gives an IBE and setting $n = l$ as in Waters' scheme in (5), we obtain $\epsilon_{ibe}^2 \leq 8nq\epsilon_{dbdh}$. Bellare and Ristenpart (2009) obtained essentially the same bound. This bound should be contrasted with the bound $\epsilon_{ibe} \leq 16nq\epsilon_{dbdh}$ obtained from Theorem 1. As mentioned in Bellare and Ristenpart (2009), the later bound is generally better and the bound in Bellare and Ristenpart (2009) improves upon the later bound in certain specific cases.

If $h \geq 2$, the bound obtained from Theorem 2 becomes much less attractive compared to the bound obtained from Theorem 1. In view of this, below we analyse in details only the bound in Theorem 1.

IBE. The special case of $h = 1$ leads to the security statement for an IBE. In this case,

$$\delta(n, l, 1, q) = 8lq2^{n/l}. \quad (6)$$

Further, if we choose $l = n$ (which gives us Waters' scheme), then $\delta(n, n, 1, q) = 16nq$. This is the least degradation that is obtained from Theorem 1. The trade-off is that in this case the size of the public parameter is quite large. By choosing suitable values of $l < n$, it is possible to reduce the size of the public parameters with a consequent increase in the value of the degradation. Below we discuss this in more details.

The effect of security degradation has been studied (Galindo, 2004; Bellare and Ristenpart, 2009) using the so-called *work factor* which is the time by advantage ratio. In our setting, for the IBE, we define $\mathsf{WF}_{\mathrm{IBE}} = t/\epsilon_{ibe}$ and for DBDH, define $\mathsf{WF}_{\mathrm{DBDH}} = t'/\epsilon_{dbdh}$. The bound in Theorem 1 says that $\epsilon_{ibe} \leq \delta\epsilon_{dbdh}$. In the following analysis, we assume that this relation holds with equality.

$$\begin{aligned}\mathsf{WF}_{\mathrm{DBDH}} = \frac{t'}{\epsilon_{dbdh}} &= \frac{\delta(n,l,1,q)(t+t_{sim})}{\epsilon_{ibe}} \\ &\geq \frac{\delta(n,l,1,q)\max(t,t_{sim})}{\epsilon_{ibe}} \\ &\geq \delta(n,l,1,q)\mathsf{WF}_{\mathrm{IBE}}.\end{aligned}$$

The security parameter κ for the IBE scheme is defined to be $\kappa = \lg(t/\epsilon_{ibe})$ where lg is the logarithm to base two. So,

$$\lg \mathsf{WF}_{\mathrm{DBDH}} \geq \kappa + \lg \delta(n,l,1,q). \tag{7}$$

In the scheme, identities are n bits long, but, in actual practice, identities are arbitrary length strings and will be hashed into n-bit strings using a collision-resistant hash function. For κ-bit security of such hash function, we require $n = 2\kappa$ (or more).

Suppose we fix a value of κ. Then as l decreases from n to 1, the value of $\lg \mathsf{WF}_{\mathrm{DBDH}}$ increases. In other words, if $l = n$, then we obtain the minimum work factor for the DBDH problem and as l decreases, the value of this work factor also increases. The DBDH work factor quantifies the difficulty of solving the DBDH problem and so it is of interest to measure the amount by which the DBDH work factor increases as l decreases. This is clearly quantified by $\zeta(n,l) \triangleq \lg \delta(n,l,1,q) - \lg \delta(n,n,1,q)$. Using the value of $\delta(n,l,1,q)$ from (6), we have

$$\zeta(n,l) = \frac{n}{l} - \lg \frac{n}{l} - 1. \tag{8}$$

As an example, if the ratio $n/l = 16$, then $\zeta(n,l) = 11$, i.e., the logarithm of the DBDH work factor is larger by 11 bits compared to the case $l = n$. In Table 1, we compare the sizes of the public parameters and degradations for $\kappa = 80$ and 128. These two values of the security parameter are of the most practical interest. In this table, the values of $\zeta(n,l) = 0$ correspond to Waters' scheme. We see that for $\kappa = 80$, the number of elements of $\mathbb{G}_1$ required in the public parameters can be reduced from 161 in Waters' scheme to 21, 11 and 6 respectively with associated increase in DBDH work factor of 4, 11 and 26 bits. Similarly, for $\kappa = 128$, the number of $\mathbb{G}_1$ elements can be reduced from 257 in Waters' scheme to 33, 17 and 9 respectively with a similar associated increase in DBDH work factor. In both cases, the decrease in the size of public parameters is quite substantial and leads to practical schemes.

The effect of increase in DBDH work factor can be offset by working in larger size groups. Heuristics estimates of such sizes can be obtained from analysis of cryptographic key sizes in Lenstra and Verheul (2001). In our conference paper (Chatterjee and Sarkar, 2005) we have provided such estimates. However, these estimates require working with the asymptotic expression for the runtime of the number field sieve algorithm for finding discrete logarithms over a finite field. They can hardly be considered to be exact and in particular, it is not clear whether these expressions can accurately capture small differences in the work factors. So, one may quite reasonably choose to ignore the small increases in the DBDH work factors of 4 or 11 bits. On the other hand, an increase of 26 bits may not be considered to be small and may require an increase in the group sizes.

Table 1 Comparison of the size of public parameters and security degradation

κ	n	l	PP	$\zeta(n,l)$	κ	n	l	PP	$\zeta(n,l)$
80	160	160	(161,1,1)	0	128	256	256	(257,1,1)	0
80	160	20	(21,1,1)	4	128	256	32	(33,1,1)	4
80	160	10	(11,1,1)	11	128	256	16	(17,1,1)	11
80	160	5	(6,1,1)	26	128	256	8	(9,1,1)	26

Waters (2009) describes an IBE scheme where the size of the public parameters is constant and does not depend on the security parameter. The security of this scheme is based on the hardnesses of both the decision linear (DLIN) problem and the DBDH problem. Security degradation is not clearly stated in Waters (2009), but, on going through the proofs, a degradation by a factor of q on the advantage of DLIN is seen. A summary of the time and space requirement of this scheme is given in Table 2 along with a comparison to the IBE scheme that is obtained from HIBE-1 of Section 3.1.

For practical situations, choosing a small value for l (for example, for 80-bit security one may choose $l = 5, 10$ or 20 and for 128-bit security, one may choose $l = 8, 16$ or 32) the size of the public parameters in the scheme described here is smaller or comparable to the size of the public parameters in Waters (2009). In every other aspect, the scheme described here is better than that of Waters (2009).

The description of the IBE scheme in Waters (2009) is given in terms of symmetric pairing and it is mentioned in the passing that use of asymmetric pairings will reduce the number of public parameters by three. This will also reduce the time for encryption, decryption and key generation. However, even such a reduced variation will still be quite inefficient compared to the schemes given here. Decryption, for example, will require six pairings which will be significantly slower than the single computation of the ratio of two pairings required for decryption in the IBE versions of HIBE-1 or HIBE-2 (see Table 2).

HIBE. In the scheme in Section 3.1, if we have $h > 1$, then we obtain a proper HIBE scheme. As mentioned earlier, the bound in Theorem 1 degrades exponentially with h and hence the scheme described here is only useful for HIBE schemes of small depths (either 2 or 3). Gentry and Halevi (2009) and later Waters (2009) described HIBE schemes which avoid this exponential security loss. The scheme in Gentry and Halevi (2009) is very complicated and is also based on a very complicated security assumption. In contrast, the scheme in Waters (2009) is simpler and based on DLIN and DBDH assumptions. Though it is not explicitly mentioned, the proofs show a degradation by a factor of q on the DLIN advantage and this is independent of the depth of the HIBE. If the depth of the HIBE scheme is required to be long, then clearly the scheme in Waters (2009) is better than the scheme described here.

On the other hand, if one is interested in small depth HIBE schemes, then the present construction offers significant efficiency improvements over the scheme in Waters (2009). Table 3 provides a comparative study of the different parameters of the two schemes. As in the case of IBE, for practical situations of 80-bit or 128-bit security levels, one may choose a suitable value of l ($l = 5, 10$ or 20 for 80-bit security and $l = 8, 16$ or 32 for 128-bit security) and then the size of the public parameters of the present scheme is comparable to that of Waters (2009). In all other aspects, the present scheme is better than that in Waters (2009).

Waters (2005) had suggested a construction of a HIBE scheme obtained by extending his IBE scheme. The number of public parameters for this scheme is $(n+1)h$. In contrast, by setting $l = n$ we obtain $n + h$ public parameters without any additional security degradation compared to Waters (2005) scheme.

4 Proofs of Theorems 1 and 2

We provide the proof for HIBE-1 – essentially the same technique applies for HIBE-2. Our goal is to show that HIBE-1 is (ϵ_{hibe}, t, q)-CPA secure. In the game sequence style of proofs, we start with the adversarial game defining the CPA-security of the scheme against an adversary $\mathcal{A}$ and then obtain a sequence of games as usual. In each of the games, the simulator chooses a bit γ and the adversary makes a guess γ'. By X_i we will denote the event that the bit γ is equal to the bit γ' in the i^{th} game.

The basic structure of the games in both the proofs of Theorems 1 and 2 are similar. Analysis of the games are different for the two proofs. The proof of Theorem 1 requires an additional 'artificial abort' step. As a result, the probability analysis of the two proofs are different. In our descriptions of the two proofs, we first describe the common structure of the games that are used in both the proofs. After doing this, we separately present the two different probability analyses.

4.1 Structure of the games

Game 0. This is the usual adversarial game used in defining CPA-secure HIBE. We assume that the adversary's runtime is t and it makes q key extraction queries. Also, we assume that the adversary maximizes the advantage among all adversaries with similar resources. Thus, we have $\epsilon_{hibe} = \left|\Pr[X_0] - \frac{1}{2}\right|$.

Game 1. Suppose P_1 (resp. P_2) is the fixed generator of $\mathbb{G}_1$ (resp. $\mathbb{G}_2$). Consider a tuple of the form: $(aP_1, aP_2, bP_1, cP_1, cP_2, e(P_1,P_2)^{abc})$ where a, b and c are chosen uniformly and independently at random from $\mathbb{Z}_p$. The simulator is assumed to know the values a, b and c. But, the simulator can setup the scheme as well as answer certain private key queries without the knowledge of these values. Also, for certain challenge identities it can generate the challenge ciphertext without the knowledge of a, b and c. In the following, we show how this can be done. If the simulator cannot answer a key extraction query or generate a challenge without using the knowledge of a, b and c, it sets a flag flg to one. The value of flg is initially set to zero.

Note that the simulator is always able to answer the adversary (with or without using a, b and c). The adversary is provided with proper replies to all its queries and is also provided the proper challenge ciphertext. Thus, irrespective of whether flg is set to one, the adversary's view in Game 1 is same as that in Game 0. Hence, we have $\Pr[X_0] = \Pr[X_1]$.

We next show how to setup the scheme and answer the queries based on the tuple

$$(aP_1, aP_2, bP_1, cP_1, cP_2, Z = e(P_1,P_2)^{abc}).$$

Set-up. Choose $x'_1, \ldots, x'_h$ and $x_1, \ldots, x_l$ randomly from $\mathbb{Z}_m$; $y'_1, \ldots, y'_h$ and $y_1, \ldots, y_l$ randomly from $\mathbb{Z}_p$. Choose $k_1, \ldots, k_h$ randomly from $\{0, \ldots, \mu_l\}$.

$$\left.\begin{array}{ll} U'_j = (p - mk_j + x'_j)bP_1 + y'_jP_1, & 1 \le j \le h; \\ U_i = x_ibP_1 + y_iP_1, & 1 \le i \le l. \end{array}\right\} \tag{9}$$

Set the public parameters of HIBE to be $(P_2, Q_2 = aP_2, R_1 = bP_1, U'_1, \ldots, U'_h, U_1, \ldots, U_l)$. The master secret is $aR_1 = abP_1$. In its attack, $\mathcal{A}$ will make some queries, which have to be properly answered by the simulator.

For $1 \le j \le h$, we define several functions. Let $v = (v_1, \ldots, v_l)$ where each v_i is an n/l-bit string considered to be an integer from the set $\{0, \ldots, 2^{n/l} - 1\}$. We define

$$\left.\begin{array}{l} F_j(v) = p - mk_j + x'_j + \sum_{i=1}^{l} x_iv_i \\ J_j(v) = y'_j + \sum_{i=1}^{l} y_iv_i \\ L_j(v) = x'_j + \sum_{i=1}^{l} x_iv_i \pmod m \\ K_j(v) = \begin{cases} 0 \text{ if } L_j(v) = 0 \\ 1 \text{ otherwise.} \end{cases} \end{array}\right\} \tag{10}$$

Recall that we have assumed $m(1 + \mu_l) < p$ (see the statement of Theorems 1 and 2). Let $F_{\min}$ and $F_{\max}$ be the minimum and maximum values of $F_j(v)$. $F_{\min}$ is achieved when k_j is maximum and x'_j and the x_i's are all zero. Thus, $F_{\min} = p - m\mu_l$. We have $m\mu_l < m(1 + \mu_l)$ and by assumption $m(1 + \mu_l) < p$. Hence, $F_{\min} > 0$. Again $F_{\max}$ is achieved when $k_j = 0$ and x'_j and the x_i's and v_i's are equal to their respective maximum values. We get $F_{\max} < p + m(1 + l(2^{n/l} - 1)) = p + m(1 + \mu_l) = p + m(1 + \mu_l) < 2p$. Thus, we have $0 < F_{\min} \le F_j(v) \le F_{\max} < 2p$. Consequently, $F_j(v) \equiv 0 \bmod p$ if and only if $F_j(v) = p$ which holds if and only if $-mk_j + x'_j + \sum_{i=1}^{l} x_iv_i = 0$.

Now we describe how the queries made by $\mathcal{A}$ are answered. The queries can be made in both Phases 1 and 2 of the adversarial game (subject to the usual restrictions). The manner in which they are answered by the simulator is the same in both the phases.

Key extraction query. Suppose $\mathcal{A}$ makes a key extraction query on the identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$. Choose random $r_1, \ldots, r_j$ from $\mathbb{Z}_p$. Suppose there is a u with $1 \le u \le j$ such that $K_u(\mathsf{v}_u) = 1$. Otherwise set flg to one. In the second case, the simulator uses the value of a to return the proper decryption key $d_{\mathsf{v}} = (abP_1 + \sum_{i=1}^{j} r_i V_i^{(1)}, r_1 V_1^{(1)}, \ldots, r_j V_j^{(1)})$. In the first case, the simulator constructs a decryption key in the following manner.

$$\left.\begin{array}{l} d_{0|u} = -\frac{J_u(\mathsf{v}_u)}{F_u(\mathsf{v}_u)} aP_1 + r_u(F_u(\mathsf{v}_u)bP_1 + J_u(\mathsf{v}_u)P_1) \\ d_u = \frac{-1}{F_u(\mathsf{v}_u)} aP_2 + r_u P_2 \\ d_k = r_k P_2 \text{ for } k = \{1, \ldots, u-1, u+1, \ldots, j\} \\ d_0 = d_{0|u} + \sum_{k \in \{1,\ldots,j\}\setminus\{u\}} r_k V_k^{(1)} \end{array}\right\} \quad (11)$$

The key $d_{\mathsf{v}} = (d_0, d_1, \ldots, d_j)$ is provided to $\mathcal{A}$. The quantity d_{v} is a proper private key corresponding to the identity v. This technique for simulating private key generation was introduced by Boneh and Boyen (2004a) and is crucial to the security reduction. For $1 \le i \le j$ and $j \ne u$, clearly d_i is properly constructed. So, we only need to show that d_0 and d_u are also properly constructed. For this it is enough to show that $d_{0|u}$ is equal to $aP_2 + \tilde{r}_u V_u^{(2)}$ and $d_u = \tilde{r}_u P$ for some uniform random $\tilde{r}_u$ which is independent of the other rs.

$$\begin{aligned} d_{0|u} &= abP_1 - a\frac{F_u(\mathsf{v}_u)}{F_u(\mathsf{v}_u)} bP_1 - \frac{J_u(\mathsf{v}_u)}{F_u(\mathsf{v}_u)} aP_1 \\ &\quad + r_u(F_u(\mathsf{v}_u)bP_1 + J_u(\mathsf{v}_u)P_1) \\ &= abP_1 - \frac{a}{F_u(\mathsf{v}_u)}(F_u(\mathsf{v}_u)bP_1 + J_u(\mathsf{v}_u)P_1) \\ &\quad + r_u(F_u(\mathsf{v}_u)bP_1 + J_u(\mathsf{v}_u)P_1) \\ &= abP_1 + \left(r_u - \frac{a}{F_u(\mathsf{v}_u)}\right)(F_u(\mathsf{v}_u)bP_1 + J_u(\mathsf{v}_u)P_1) \\ &= abP_1 + \tilde{r}_u\left(\left(p - mk_u + x'_u + \sum_{i=1}^{l} x_i \mathsf{v}_{u,i}\right) bP_1 \right. \\ &\quad \left. + \left(y'_u + \sum_{i=1}^{l} y_i \mathsf{v}_{u,i}\right) P_1\right) \\ &= abP_1 + \tilde{r}_u\big((p - mk_u + x'_u)bP_1 + y'_u P_1. \\ &\quad + \sum_{i=1}^{l} \mathsf{v}_{u,i}\,(x_i bP_1 + y_i P_1)\big) \\ &= abP_1 + \tilde{r}_u\left(U'_u + \sum_{i=1}^{l} \mathsf{v}_{u,i} U_i\right) \\ &= aR_1 + \tilde{r}_u V_u^{(1)}(\mathsf{v}_u). \end{aligned}$$

Here $\tilde{r}_u = r - a/F_u(\mathsf{v}_u)$ and is a uniform random value over $\mathbb{Z}_p$ and is independent of the other rs as required. A simpler calculation shows that $d_u = \tilde{r}_u P_2$.

Challenge. Let the challenge identity be $\mathsf{v}^* = (\mathsf{v}_1^*, \ldots, \mathsf{v}_{h^*}^*)$, $1 \le h^* \le h$ and the (equal length) messages be M_0 and M_1. Choose a random bit γ. We need to have $F_k(\mathsf{v}_k^*) \equiv 0 \bmod p$ for all $1 \le k \le h^*$. If this condition does not hold, then set flg to one. In the second case, the simulator uses the value of c to provide a proper encryption of M_γ to $\mathcal{A}$ by computing $(M_\gamma \times e(R_1, Q_2)^c, cP_2, cV_1, \ldots, cV_{h^*})$. In the first case, it constructs a proper encryption of M_γ in the following manner.

$$\begin{aligned} (M_\gamma \times Z, C_1 = cP_2, B_1 = J_1(\mathsf{v}_1^*)cP_1, \ldots, B_{h^*} \\ = J_{h^*}(\mathsf{v}_{h^*}^*)cP_1)). \end{aligned}$$

We require B_k to be equal to $cV_k^{(1)}(\mathsf{v}_k^*)$ for $1 \le k \le h^*$. Using the definition of U'_j and the U_i's we obtain,

$$cV_k^{(1)}(\mathsf{v}_k^*) = c(F_k(\mathsf{v}_k^*)bP_1 + J_k(\mathsf{v}_k^*)P_1) = J_k(\mathsf{v}_k^*)cP_1.$$

Here we use the fact, $F_k(\mathsf{v}_k^*) \equiv 0 \bmod p$. Hence, the quantities $B_1, \ldots, B_{h^*}$ are properly formed.

Guess. The adversary outputs a guess γ' of γ.

Game 2. This is a modification of Game 1 whereby the Z in Game 1 is now chosen to be a random element of $\mathbb{G}_2$. This Z is used to mask the message M_γ in the challenge ciphertext. Since Z is random, the first component of the challenge ciphertext is a random element of $\mathbb{G}_T$ and provides no information to the adversary about γ. Thus, $\Pr[X_2] = \frac{1}{2}$.

Let flg_i denote the random variable flg in Game i, $i = 1, 2$. Denote by $\mathsf{v}^{(1)}, \ldots, \mathsf{v}^{(q)}$ the identities in the q key extraction queries and v^* is the challenge identity. Let $\vec{\mathsf{V}} = (\mathsf{v}^*, \mathsf{v}^{(1)}, \ldots, \mathsf{v}^{(q)})$.

Let $\vec{X}$ be the tuple of random variables consisting of the x's and the x''s used during set-up. Let $\vec{Z}$ be the tuple of random variables consisting of the adversary's private random bits; the k's, the y's and the y''s used during set-up; and the r's used in answering the key extraction queries. Then $\vec{X}$ and $\vec{Z}$ are independent random variables. A specific value of $\vec{X}$ will be denoted by $\vec{x}$; a specific value of $\vec{Z}$ will be denoted by $\vec{z}$; and a specific value of $\vec{\mathsf{V}}$ will be denoted by $\vec{\mathsf{v}}$.

Proposition 1. For any fixed $\vec{\mathsf{v}}$, let $\lambda(\vec{\mathsf{v}}) \stackrel{\Delta}{=} \Pr[\mathsf{flg}_i(\vec{X}, \vec{\mathsf{v}}) = 0]$. Then $\lambda^- \le \lambda(\vec{\mathsf{v}}) \le \lambda^+$ where

$$\lambda^- = \left(1 - \frac{q}{m}\right)\lambda^+ \text{ and } \lambda^+ = \frac{1}{(m(\mu_l + 1))^h}.$$

The proof of Proposition 1 is given in Section 4.5.

4.2 *Analysis for Theorem 1*

Recall that in this case $m = \max(2q, 2^{n/l})$ and so $\lambda^- \ge 1/(2(m(\mu_l + 1))^h)$. We set $\lambda \stackrel{\Delta}{=} 1/(2(m(\mu_l + 1))^h)$.

We show that it is possible to obtain an algorithm $\mathcal{B}$ for DBDH in $(p, \mathbb{G}_1 = \langle P_1 \rangle, \mathbb{G}_2 = \langle P_2 \rangle, \mathbb{G}_T, e)$ by extending Games 1 and 2. The extensions of both the games are same and is described as follows. $\mathcal{B}$ takes as input a tuple $(aP_1, aP_2, bP_1, cP_1, cP_2, Z)$ and sets up the HIBE scheme as in Game 1. Also, the key extraction queries are answered and the challenge ciphertext is generated as in Game 1. If at any stage, flg is set to one, then $\mathcal{B}$ outputs a random bit and aborts. At the end of the game, the adversary outputs the guess γ'.

Artificial abort. This technique was introduced by Waters (2005). The probability that $\mathcal{B}$ aborts in the query or challenge phases depends on the adversary's input. This probability depends on transcript of the adversary, i.e., the identities queried by the adversary and also the challenge identity. The goal of the artificial abort step is to ensure that $\mathcal{B}$ aborts with more or less the same probability irrespective of the transcript of the adversary.

Suppose that flg_i remains 0 during the simulation which denotes the fact that there is no need to abort at the end of the game. At this point, the values of the random variables $\vec{X}$, $\vec{Z}$ and $\vec{V}$ have been fixed. Let $\vec{z}$ and $\vec{v}$ be the values of $\vec{Z}$ and $\vec{V}$ respectively. Note that fixing $\vec{v}$ fixes the adversaries queries and the challenge identity. Further, given these values $\vec{z}$ and $\vec{v}$, $\mathcal{B}$ can sample a new uniform random value for $\vec{X}$ and evaluate whether it would have needed to abort for this sampled value.

$\mathcal{B}$ now goes through an additional abort step. It does the above sampling of $\vec{X}$ a total of

$$O(\epsilon_{hibe}^{-2}\lambda^{-1}\ln(\epsilon_{hibe}^{-1}\lambda^{-1}))$$

times and for each sample decides whether it would have aborted or not. Based on the sum total of this decision, it decides whether to abort or not. The exact details of this strategy is explained in Section 4.3.

Output of $\mathcal{B}$ in Games 1 and 2. If $\mathcal{B}$ has not aborted both at the end of game and also in the artificial abort step, then $\mathcal{B}$ outputs 1 if $\gamma = \gamma'$; else 0.

The time taken by $\mathcal{B}$ in either Game 1 or 2 is easily seen to be as stated in the statement of Theorem 1.

If Z is real, then the adversary is playing Game 1 and if Z is random, then the adversary is playing Game 2. Let Y_i be the event that the simulator outputs 1 in Game i, $i = 1, 2$. Then, we have

$$|\mathsf{Pr}[Y_1] - \mathsf{Pr}[Y_2]| \le \epsilon_{dbdh}.$$

Let ab_i be the event that the simulator aborts in Game i, $i = 1, 2$. This includes both usual abort (i.e., due to $\mathsf{flg}_i = 1$) and artificial abort.

$$\begin{aligned}
&\mathsf{Pr}[Y_i] \\
&= \mathsf{Pr}[Y_i \wedge (\mathsf{ab}_i \vee \overline{\mathsf{ab}_i})] \\
&= \mathsf{Pr}[(Y_i \wedge \mathsf{ab}_i) \vee (Y_i \wedge \overline{\mathsf{ab}_i})] \\
&= \mathsf{Pr}[Y_i \wedge \mathsf{ab}_i] + \mathsf{Pr}[Y_i \wedge \overline{\mathsf{ab}_i}] \\
&= \mathsf{Pr}[Y_i \mid \mathsf{ab}_i]\mathsf{Pr}[\mathsf{ab}_i] + \mathsf{Pr}[Y_i \mid \overline{\mathsf{ab}_i}]\mathsf{Pr}[\overline{\mathsf{ab}_i}] \\
&= \frac{1}{2}(1 - \mathsf{Pr}[\overline{\mathsf{ab}_i}]) + \mathsf{Pr}[X_i \mid \overline{\mathsf{ab}_i}]\mathsf{Pr}[\overline{\mathsf{ab}_i}] \qquad (12) \\
&= \frac{1}{2}(1 - \mathsf{Pr}[\overline{\mathsf{ab}_i} \wedge (X_i \vee \overline{X_i})]) + \mathsf{Pr}[X_i \wedge \overline{\mathsf{ab}_i}] \\
&= \frac{1}{2} + \frac{1}{2}\left(\mathsf{Pr}[\overline{\mathsf{ab}_i}|X_i]\mathsf{Pr}[X_i] - \mathsf{Pr}[\overline{\mathsf{ab}_i}|\overline{X_i}]\mathsf{Pr}[\overline{X_i}]\right) \qquad (13)
\end{aligned}$$

Table 2 Comparison to Waters (2009) IBE scheme

Scheme	*PP*	*msk*	*sec key*	*cpr txt*	*key gen*	*enc*	*dec*
HIBE-1 $h=1$	$(l+1$,1,1,0)	(1,0,0,0)	(1,1,0,0)	(1,1,1,0)	1[SM$_1$]+1[SM$_2$] +1$\left[\mathrm{H}_{n,l}^{(1)}\right]$	1[SM$_1$]+1[SM$_2$] +1[E]+1$\left[\mathrm{H}_{n,l}^{(1)}\right]$	1[P$_2$]
HIBE-2 $h=1$	$(l+2, l+2, 1, 0)$	(0,1,0,0)	(0,2,0,0)	(2,0,1,0)	2[SM$_2$]+1$\left[\mathrm{H}_{n,l}^{(2)}\right]$	2[SM$_1$] +1[E]+1$\left[\mathrm{H}_{n,l}^{(1)}\right]$	1[P$_2$]
Waters (2009)	((12, 1, 0))	((6, 0, 0))	((8, 0, 1))	((10, 0, 1))	12[SM]	14[SM] + 1[SE]	9[P]+2[SM]

Note: Waters (2009) scheme uses symmetric pairings.

Table 3 Comparison of the HIBE scheme in Waters (2009) to HIBE-1 and HIBE-2

Scheme	*Parameter sizes*				*Times*		
	PP	*msk*	*pvt key*	*cpr txt*	*key gen*	*enc*	*dec*
HIBE-1	$(h+l, 1, 1, 0)$	$(1, 0, 0, 0)$	$(1, j, 0, 0)$	$(j, 1, 1, 0)$	$1\left[\mathrm{H}_{n,l}^{(1)}\right]$ $+1[\mathrm{SM}_1] + 1[\mathrm{SM}_2]$	$j\left[\mathrm{H}_{n,l}^{(1)}\right] + 1[\mathrm{E}]$ $+j[\mathrm{SM}_1] + 1[\mathrm{SM}_2]$	$1[\mathrm{P}_{j+1}]$
HIBE-2	$(h+l+1,$ $h+l+1, 1, 0)$	$(0, 1, 0, 0)$	$(0, j+1, 0, 0)$	$(j+1, 0, 1, 0)$	$1\left[\mathrm{H}_{n,l}^{(2)}\right] + 2[\mathrm{SM}_2]$	$j\left[\mathrm{H}_{n,l}^{(1)}\right] + 1[\mathrm{E}]$ $+(j+1)[\mathrm{SM}_1]$	$1[\mathrm{P}_{j+1}]$
Waters (2009)	$((13+2h$,1,0))	((3,0,0))	$((7+2j$,0,$j))$	$((7+j$,0,$j))$	$(9+4j)$[SM]	$(11+3j)$[SM] +1[SE]	$(7+2j)$[SP$_1$] +$2j$[SM]

Notes: The quantity j refers to the number of components in the identity tuple for which the computation is carried out. The time for key generation shows only the time required to generate a key for the j^{th} level from a key for the $(j-1)^{\text{st}}$ level. Also, the comparison is meaningful only for $h \le 3$.

To proceed further, we need bounds on $\Pr[\overline{\mathsf{ab}_i}|X_i]$ and $\Pr[\overline{\mathsf{ab}_i}|\overline{X_i}]$. Recall that X_i is the event $\gamma = \gamma'$ in Game i. From (29) (proved later in Section 4.3), we obtain

$$\lambda - \frac{\lambda\epsilon}{2} \leq \Pr[\overline{\mathsf{ab}_i}|X_i], \Pr[\overline{\mathsf{ab}_i}|\overline{X_i}] \leq \lambda + \frac{\lambda\epsilon}{2}. \tag{14}$$

Here $\epsilon = \epsilon_{hibe}$. Now we need to do some manipulations with inequalities and for convenience we set $A_i = \Pr[\overline{\mathsf{ab}_i}|X_i]$, $B_i = \Pr[X_i]$ and $C_i = \Pr[\overline{\mathsf{ab}_i}|\overline{X_i}]$ and $D = \Pr[Y_1] - \Pr[Y_2]$. We have from (14)

$$\lambda - \frac{\lambda\epsilon}{2} \leq A_i, C_i \leq \lambda + \frac{\lambda\epsilon}{2}.$$

Also, using (13)

$$\begin{aligned} 2D = {} & (A_1B_1 - C_1(1 - B_1)) \\ & - (A_2B_2 - C_2(1 - B_2)). \end{aligned} \tag{15}$$

Since both B_1 and $(1 - B_1)$ are non-negative, we have

$$\begin{array}{rcl} B_i(\lambda - \frac{\lambda\epsilon}{2}) \leq & A_iB_i & \leq B_i(\lambda + \frac{\lambda\epsilon}{2}) \\ (1 - B_i)(-\lambda - \frac{\lambda\epsilon}{2}) \leq & -C_i(1 - B_i) & \leq (1 - B_i)(-\lambda + \frac{\lambda\epsilon}{2}). \end{array}$$

Hence,

$$\begin{aligned} \lambda(2B_i - 1) - \frac{\lambda\epsilon}{2} & \leq A_iB_i - C_i(1 - B_i) \\ & \leq \lambda(2B_i - 1) + \frac{\lambda\epsilon}{2}. \end{aligned} \tag{16}$$

Putting $i = 1$ in (16), we obtain

$$\begin{aligned} \lambda(2B_1 - 1) - \frac{\lambda\epsilon}{2} & \leq A_1B_1 - C_1(1 - B_1) \\ & \leq \lambda(2B_1 - 1) + \frac{\lambda\epsilon}{2}. \end{aligned} \tag{17}$$

Multiplying (16) by -1 and putting $i = 2$ we obtain

$$\begin{aligned} -\lambda(2B_2 - 1) - \frac{\lambda\epsilon}{2} & \leq -(A_2B_2 - C_2(1 - B_2)) \\ & \leq -\lambda(2B_2 - 1) + \frac{\lambda\epsilon}{2}. \end{aligned} \tag{18}$$

Combining (15), (17) and (18) we get

$$2\lambda(B_1 - B_2) - \lambda\epsilon \leq 2D \leq 2\lambda(B_1 - B_2) + \lambda\epsilon. \tag{19}$$

This shows that $|\lambda(B_1 - B_2) - D| \leq \frac{\lambda\epsilon}{2}$. Now $|\lambda(B_1 - B_2)| - |D| \leq |\lambda(B_1 - B_2) - D| \leq \frac{\lambda\epsilon}{2}$. Note that $|D| = |\Pr[Y_1] - \Pr[Y_2]| \leq \epsilon_{dbdh}$ and recalling the values of B_1 and B_2, we have

$$|\Pr[X_1] - \Pr[X_2]| \leq \frac{\epsilon_{dbdh}}{\lambda} + \frac{\epsilon_{hibe}}{2}. \tag{20}$$

This completes the proof of the claim. □

Now we can complete the proof in the following manner.

$$\begin{aligned} \epsilon_{hibe} & = \left|\Pr[X_0] - \frac{1}{2}\right| \\ & \leq |\Pr[X_0] - \Pr[X_2]| \\ & \leq |\Pr[X_0] - \Pr[X_1]| + |\Pr[X_1] - \Pr[X_2]| \\ & \leq \frac{\epsilon_{hibe}}{2} + \frac{\epsilon_{dbdh}}{\lambda}. \end{aligned}$$

Rearranging the inequality gives the desired result. This completes the proof of Theorem 1. □

4.3 Artificial abort: details and analysis

The entire technique of artificial abort can be explained in terms of elementary probability theory without reference to security proofs. We have taken this approach. After outlining the necessary abstraction, we obtain the relevant bounds. Then, we go back to the security proof and explain how the technique fits within that scenario.

Let $\mathsf{Ber}(\eta)$ be the Bernoulli distribution with probability of success η. Suppose $X_1, \ldots, X_k$ are independent random variables each following $\mathsf{Ber}(\eta)$. Let $\eta' = (\sum_{i=1}^{k} X_i)/k$. Then for $0 < \delta \leq 2e - 1$, we have using Chernoff bound (see Motwani and Raghavan (1995))

$$\Pr[|\eta' - \eta| \geq \delta\eta] \leq 2 \times \exp\left(\frac{-k\eta\delta^2}{4}\right). \tag{21}$$

Let λ be a lower bound on η, i.e., $\lambda \leq \eta$ and ϵ be such that $0 < \epsilon < 1$. We want to ensure $2 \times \exp\left(\frac{-k\eta\delta^2}{4}\right) \leq \frac{\lambda\epsilon}{8}$ and $\delta = \epsilon/8$. This gives us the condition: $2 \times \exp((-k\eta\epsilon^2)/256) \leq \lambda\epsilon/8$. This condition holds when

$$k \geq \left(\frac{256}{\lambda\epsilon^2} \ln \frac{16}{\lambda\epsilon}\right). \tag{22}$$

If we consider η' to be an approximation of η then, for k satisfying (22), with high probability η' is a good approximation of η. Note that the lower bound on k in (22) gives rise to the expression in the statement of Theorem 1. The summary of all this is that for k satisfying (22) the following inequality holds.

$$\Pr[|\eta' - \eta| \geq \delta\eta] \leq \frac{\lambda\epsilon}{8}. \tag{23}$$

We now present an abstract description of the artificial abort technique. Let X (resp. Y) be a random variable which varies over a finite set Γ (resp. Σ). Suppose that the random variable X follows the uniform distribution, while the distribution of Y is unknown. Also, let X and Y be independent. Let f be a function $f : \Gamma \times \Sigma \to \{0, 1\}$ and consider the following procedure.

Procedure-A

1. Choose x from Γ according to the uniform distribution.
2. A value y is obtained from Σ according to some unknown distribution.
3. Output $f(x, y)$.

X (resp. Y) is the random variable representing x (resp. y). The probability of outputting 1 in the above game depends on the distribution of Y. Let η_y be the probability that $f(X, y) = 1$, i.e., $\eta_y = \mathsf{Pr}[f(X, y) = 1] = \mathsf{Pr}[f(X, Y) = 1|Y = y]$ (since X and Y are independent). Since the distribution of Y is not known, the value of η_y is also not known. Let λ be a known lower bound on η_y, i.e., $\lambda \le \eta_y$ for all $y \in \Sigma$.

We want to augment Procedure-A, such that the probability of outputting 1 remains more or less the same irrespective of the choice of y. This is done in the following manner. By $\mathsf{Ber}(p)$ we mean the Bernoulli experiment where 1 is produced with probability p and 0 is produced with probability $(1 - p)$.

Procedure-B

1. Choose x from Γ according to the uniform distribution.
2. A value y is obtained from Σ according to some unknown distribution;
3. if $f(x, y) = 1$, then
4. choose $x^{(1)}, \ldots, x^{(k)}$ uniformly at random from Γ;
5. define $\eta' = \dfrac{\sum_{i=1}^{k} f(x^{(i)}, y)}{k}$;
6. if $\eta' \ge \lambda$, then perform $\mathsf{Ber}(\lambda/\eta')$;
7. else output 1;
8. else output 0.

The quantity η' computed in Step 5 is an estimate of η_y, for the y given in Step 2. To understand what is happening, first suppose that the estimate η' of η_y is exact. Since λ is a lower bound on η_y, the condition in Step 6 will be true and hence the above procedure will output 1 with probability $\eta \times \lambda/\eta = \lambda$. In other words, the above procedure will output 1 with probability λ irrespective of the value of y which is what we want. Now the estimate η' will not be exact and consequently the probability the Procedure-B outputs 1 will be between two bounds as we outline below.

Analysis of Procedure-B. We now perform the probability analysis that Procedure-B outputs 1. In the following, we will use η to denote η_y, where y is chosen in Step 2 of Procedure-B. Let ab be the event that Procedure-B outputs 0. We are interested in the event $\overline{\mathsf{ab}}$, i.e., the event that Procedure-B outputs 1. More particularly, we are interested in the probability of $\overline{\mathsf{ab}}$ when the choice of y in Step 2 of Procedure-B is fixed.

Let $\mathbf{A}$ be the event that $|\eta' - \eta| \le \frac{\eta\epsilon}{8}$. Using standard probability arguments (for events A and B, $\Pr[A] = \Pr[A|B]\Pr[B] + \Pr[A|\overline{B}]\Pr[\overline{B}]$) it is possible to show the following.

$$\begin{aligned}\mathsf{Pr}[\overline{\mathsf{ab}}|Y = y, \mathbf{A}] - \mathsf{Pr}[\overline{\mathbf{A}}] \le \mathsf{Pr}[\overline{\mathsf{ab}}|Y = y] \\ \le \mathsf{Pr}[\overline{\mathsf{ab}}|Y = y, \mathbf{A}] + \mathsf{Pr}[\overline{\mathbf{A}}].\end{aligned} \tag{24}$$

First note that using the Chernoff bound analysis, we have from (23) $\mathsf{Pr}[\overline{\mathbf{A}}] \le \lambda\epsilon/8$. We now consider $\mathsf{Pr}[\overline{\mathsf{ab}}|Y = y, \mathbf{A}]$. There are two cases to consider.

Case 1 ($\lambda \le \eta - \frac{\eta\epsilon}{8}$). Suppose that $\mathbf{A}$ holds. Then

$$\eta - \tfrac{\eta\epsilon}{8} \le \eta' \le \eta + \tfrac{\eta\epsilon}{8}. \tag{25}$$

By the condition of this case, we have $\lambda \le \eta - \frac{\eta\epsilon}{8}$ and so $\eta' \ge \lambda$. Hence, the 'if' condition in Step 6 of Procedure-B is satisfied and therefore $\mathsf{Pr}[\overline{\mathsf{ab}}|Y = y, \mathbf{A}] = \eta\lambda/\eta'$. Using (25), it is easy to work out that for $\epsilon \le 4$,

$$\lambda\left(1 - \frac{\epsilon}{4}\right) \le \frac{\lambda}{1 + \frac{\epsilon}{8}} \le \frac{\lambda\eta}{\eta'} \le \frac{\lambda}{1 - \frac{\epsilon}{8}} \le \lambda\left(1 + \frac{\epsilon}{4}\right).$$

This combined with the bound on $\mathsf{Pr}[\overline{\mathbf{A}}]$ shows that

$$\begin{aligned}\lambda\left(1 - \frac{\epsilon}{2}\right) &\le \lambda\left(1 - \frac{\epsilon}{4}\right) - \frac{\lambda\epsilon}{8} \\ &\le \mathsf{Pr}[\overline{\mathsf{ab}}|Y = y] \le \lambda\left(1 + \frac{\epsilon}{4}\right) + \frac{\lambda\epsilon}{8} \\ &\le \lambda\left(1 + \frac{\epsilon}{2}\right).\end{aligned}$$

Put differently, we have

$$|\mathsf{Pr}[\overline{\mathsf{ab}}|Y = y] - \lambda| \le \frac{\lambda\epsilon}{2}. \tag{26}$$

Case 2 ($\lambda > \eta - \frac{\eta\epsilon}{8}$). As in Case 1, let $\mathbf{A}$ be the event $|\eta' - \eta| \le \frac{\eta\epsilon}{8}$. We identify two other events $\mathbf{A}_1$ and $\mathbf{A}_2$ as follows. $\mathbf{A}_1$ is the event $\eta - \eta\epsilon/8 \le \eta' \le \lambda$ and $\mathbf{A}_2$ is the event $\lambda \le \eta' \le \eta + \eta\epsilon/8$. Clearly, $\mathbf{A} = \mathbf{A}_1 \vee \mathbf{A}_2$ and $\mathsf{Pr}[\mathbf{A}] = \mathsf{Pr}[\mathbf{A}_1] + \mathsf{Pr}[\mathbf{A}_2]$. Now, $\mathsf{Pr}[\overline{\mathsf{ab}}|Y = y, \mathbf{A}_2] = \eta\lambda/\eta'$ and $\mathsf{Pr}[\overline{\mathsf{ab}}|Y = y, \mathbf{A}_1] = \eta$. When $\mathbf{A}_1$ occurs, $\eta' \le \lambda$ and so $\mathsf{Pr}[\overline{\mathsf{ab}}|Y = y, \mathbf{A}_1] = \eta \le \eta\lambda/\eta'$. We can write

$$\begin{aligned}&\mathsf{Pr}[\overline{\mathsf{ab}}|Y = y, \mathbf{A}] \\ &= \frac{\mathsf{Pr}[\overline{\mathsf{ab}}|Y = y, \mathbf{A}_1]\mathsf{Pr}[\mathbf{A}_1] + \mathsf{Pr}[\overline{\mathsf{ab}}|Y = y, \mathbf{A}_2]\mathsf{Pr}[\mathbf{A}_2]}{\mathsf{Pr}[\mathbf{A}]} \\ &\le \frac{\eta\lambda}{\eta'} \\ &\le \lambda + \frac{\lambda\epsilon}{4}.\end{aligned}$$

The last inequality follows from the analysis of the expression $\eta\lambda/\eta'$ when $\mathbf{A}$ occurs as done for Case 1. Then, as in Case 1, we have $\mathsf{Pr}[\overline{\mathsf{ab}}|Y = y] \le \lambda + \lambda\epsilon/2$.

We next consider the lower bound. Again, using the analysis of Case 1, we have

$$\mathsf{Pr}[\overline{\mathsf{ab}}|Y=y,\mathbf{A}_2]=\frac{\lambda\eta}{\eta'}\geq\lambda-\frac{\lambda\epsilon}{4}.$$

Also,

$$\mathsf{Pr}[\overline{\mathsf{ab}}|Y=y,\mathbf{A}_1]=\eta\geq\lambda\geq\lambda-\frac{\lambda\epsilon}{4}.$$

Using these two bounds, we obtain $\mathsf{Pr}[\overline{\mathsf{ab}}|Y=y,\mathbf{A}]\geq\lambda-\frac{\lambda\epsilon}{4}$ and consequently as in Case 1, $\mathsf{Pr}[\overline{\mathsf{ab}}|Y=y]\geq\lambda-\frac{\lambda\epsilon}{2}$. Thus, in both Cases 1 and 2, we have

$$|\mathsf{Pr}[\overline{\mathsf{ab}}|Y=y]-\lambda|\leq\frac{\lambda\epsilon}{2}. \tag{27}$$

Summary. Thus, (27) shows that the probability that Procedure-B outputs 1 is 'very close' to λ. Recall that y is an element of Σ. Let Σ' be any subset of Σ. Then

$$\begin{aligned}\mathsf{Pr}[\overline{\mathsf{ab}}|Y\in\Sigma'] &= \frac{\Pr[\overline{\mathsf{ab}},Y\in\Sigma']}{\Pr[Y\in\Sigma']}\\ &= \frac{\sum_{y\in\Sigma'}\Pr[\overline{\mathsf{ab}},Y=y]}{\Pr[Y\in\Sigma']}\\ &= \frac{\sum_{y\in\Sigma'}\Pr[\overline{\mathsf{ab}}|Y=y]\Pr[Y=y]}{\Pr[Y\in\Sigma']}.\end{aligned}$$

Using the upper and lower bounds on $\Pr[\overline{\mathsf{ab}}|Y=y]$ from (27) shows that

$$|\mathsf{Pr}[\overline{\mathsf{ab}}|Y\in\Sigma']-\lambda|\leq\frac{\lambda\epsilon}{2}. \tag{28}$$

Relation to the security game. So far, we have been analysing probabilities without any reference to the actual adversarial game. We now relate this analysis to the proof. This is seen by identifying the random variable X with the random variable $\vec{X}$ in the security game and the random variable Y with the random variable $\vec{Z}$ in the security game. The set Σ then corresponds to the set of possible values of $\vec{Z}$. We identify the set Σ_0 to be the set of all possible values of $\vec{Z}$ such that $\gamma=\gamma'$. Also, define $\Sigma_1=\Sigma\setminus\Sigma_0$.

Let Γ be the set of possible values of $\vec{X}$; define $f:\Gamma\times\Sigma\to\{0,1\}$ as follows: for $\vec{x}\in\Gamma$ and $\vec{z}\in\Sigma$, $f(\vec{x},\vec{z})$ is 1 if $\mathcal{B}$ does not abort during the simulation, i.e., flg is set to 0 at the end of the simulation. Then Procedure-A evaluates f. Procedure-B on the other hand, aborts in certain cases even when Procedure-A does not abort. This additional abort has been termed 'artificial abort' in Waters (2005).

We now follow the analysis of Procedure-B given in Section 4.3. Successively using Σ_0 and Σ_1 in place of Σ' in (28), we have,

$$\lambda-\frac{\lambda\epsilon}{2}\leq\mathsf{Pr}[\overline{\mathsf{ab}}|\gamma=\gamma'],\mathsf{Pr}[\overline{\mathsf{ab}}|\gamma\neq\gamma']\leq\lambda+\frac{\lambda\epsilon}{2}. \tag{29}$$

4.4 Analysis for Theorem 2

Bellare and Ristenpart (2009) made some crucial observations. Propositions 2 and 3 are based on the discussion given in Bellare and Ristenpart (2009).

Proposition 2.

1. $\vec{\mathsf{V}}$ is independent of $\vec{X}$
2. in the i^{th} game, the event X_i is independent of $\vec{X}$
3. the random variable flg_i is a function $\mathsf{flg}_i\triangleq\mathsf{flg}_i(\vec{X},\vec{\mathsf{V}})$.

Proof:

1. Fix any value $\vec{x}$ for $\vec{X}$. Irrespective of this value, the independent and uniform random choices of the y's and the y''s ensure that the public parameters are independent and uniformly distributed points. The independent and uniform random choices of the r's ensure that the response to any query is uniform random and independent of other random variables. Lastly, the independent and uniform randomness of c ensures that the challenge ciphertext is independent of $\vec{X}$. This is true irrespective of whether flg is set to 1 or 0.

   The adversary's queries depends on its own random choices, the distribution of the public parameters, the responses to the queries and the challenge ciphertext. By the above argument, for every fixed value of $\vec{X}$, the distributions of these random variables are the same. Hence, for every fixed value of $\vec{X}$, the probability that the adversary outputs a particular query sequence is a constant, i.e., $\Pr[\vec{\mathsf{V}}=\vec{\mathsf{v}}|\vec{X}=\vec{x}]$ does not depend on $\vec{x}$. Then it easily follows that $\vec{\mathsf{V}}$ and $\vec{X}$ are independent.
2. The event X_i is the event $\gamma=\gamma'$ in Game i. The bit γ is a uniform random bit which is independent of all other quantities. In Games 0 and 1, the adversary's output γ' is a function of its private random choices, the public parameters, the responses to the queries and the challenge ciphertext. So, as argued above, the output γ' is independent of $\vec{X}$ and hence the event $\gamma=\gamma'$ is also independent of $\vec{X}$. In Game 2, the bit γ is statistically hidden from the adversary and hence the probability of the event X_2 is $1/2$ irrespective of the value of $\vec{X}$.
3. The value of flg_i is 0 if all the F-values corresponding to the key extraction queries are non-zero and the F-value for the challenge identity is 0. From the definition of the function F, it follows that this event depends only on $\vec{X}$ and $\vec{\mathsf{V}}$ and so the random variable flg_i is a function of these two random variables. $\square$

Proposition 3.

$$\Pr[\mathsf{flg}_i = 0 | \vec{\mathsf{V}} = \vec{\mathsf{v}}, X_i] = \Pr[\mathsf{flg}_i = 0 | \vec{\mathsf{V}} = \vec{\mathsf{v}}].$$

Proof: From Proposition 2, flg_i is actually $\mathsf{flg}_i(\vec{X}, \vec{\mathsf{V}})$ and the events $\vec{\mathsf{V}} = \vec{\mathsf{v}}$ and X_i are determined by $\vec{Z}$ and is independent of $\vec{X}$. Let $f_1(\vec{Z}) \triangleq \vec{\mathsf{V}}$ and $f_2(\vec{Z}) \triangleq \gamma \oplus \gamma'$ for suitable functions f_1 and f_2. The events $\vec{\mathsf{V}} = \vec{\mathsf{v}}$ and X_i are then $f_1(\vec{Z}) = \vec{\mathsf{V}}$ and $f_2(\vec{Z}) = 0$.

$$\begin{aligned}
&\Pr[\mathsf{flg}_i = 0 | \vec{\mathsf{V}} = \vec{\mathsf{v}}, X_i] \\
&= \Pr[\mathsf{flg}_i(\vec{X}, \vec{\mathsf{V}}) = 0 | f_1(\vec{Z}) = \vec{\mathsf{v}}, f_2(\vec{Z}) = 0] \\
&= \Pr[\mathsf{flg}_i(\vec{X}, f_1(\vec{Z})) = 0 | f_1(\vec{Z}) = \vec{\mathsf{v}}, f_2(\vec{Z}) = 0] \\
&= \Pr[\mathsf{flg}_i(\vec{X}, \vec{\mathsf{v}}) = 0 | f_2(\vec{Z}) = 0] \\
&= \Pr[\mathsf{flg}_i(\vec{X}, \vec{\mathsf{v}}) = 0] \\
&\quad \text{(since } \vec{X} \text{ and } \vec{Z} \text{ are independent)} \\
&= \Pr[\mathsf{flg}_i(\vec{X}, \vec{\mathsf{V}}) = 0 | \vec{\mathsf{V}} = \vec{\mathsf{v}}] \\
&= \Pr[\mathsf{flg}_i = 0 | \vec{\mathsf{V}} = \vec{\mathsf{v}}].
\end{aligned}$$

□

We show that it is possible to obtain an algorithm $\mathcal{B}$ for DBDH in $(p, \mathbb{G}_1 = \langle P_1 \rangle, \mathbb{G}_2 = \langle P_2 \rangle, \mathbb{G}_T, e)$ by extending Games 1 and 2. The extension of both the games is same and is described as follows. $\mathcal{B}$ takes as input a tuple $(aP_1, aP_2, bP_1, cP_1, cP_2, Z)$ and sets up the HIBE scheme as in Game 1. Also, the key extraction queries are answered and the challenge ciphertext is generated as in Game 1. If at any stage, flg is set to one, then $\mathcal{B}$ outputs a random bit and aborts. At the end of the game, the adversary outputs the guess γ'. If $\mathcal{B}$ has not aborted up to this stage, then it outputs 1 if $\gamma = \gamma'$; else 0.

If Z is real, then the adversary is playing Game 1 and if Z is random, then the adversary is playing Game 2. The time taken by $\mathcal{B}$ in either Game 1 or 2 is clearly t'. Note that $\Pr[\mathsf{flg}_i(\vec{X}, \vec{\mathsf{V}}) = 0]$ is the probability that $\mathcal{B}$ does not abort during the game.

Let Y_i be the event that $\mathcal{B}$ outputs 1 in Game i, $i = 1, 2$. Then, we have

$$|\Pr[Y_1] - \Pr[Y_2]| \leq \epsilon_{dbdh}.$$

Let ab_i be the event $\mathsf{flg}_i(\vec{X}, \vec{\mathsf{v}}) = 1$, i.e., the event that the simulator aborts in Game i, $i = 1, 2$. Note that in this case there is no artificial abort.

Proposition 4. $\varepsilon_{hibe} \leq \dfrac{\varepsilon_{dbdh}}{\lambda^+} + \dfrac{q}{m}$.

$$\begin{aligned}
\Pr[Y_i] &= \Pr[Y_i \mid \mathsf{ab}_i]\Pr[\mathsf{ab}_i] + \Pr[Y_i \mid \overline{\mathsf{ab}_i}]\Pr[\overline{\mathsf{ab}_i}] \\
&= \frac{1}{2}\Pr[\mathsf{ab}_i] + \Pr[X_i \mid \overline{\mathsf{ab}_i}]\Pr[\overline{\mathsf{ab}_i}] \\
&= \frac{1}{2}\Pr[\mathsf{ab}_i] + \Pr[X_i \wedge \overline{\mathsf{ab}_i}] \\
&= \frac{1}{2}\Pr[\mathsf{ab}_i] + \Pr[X_i \wedge \mathsf{flg}_i(\vec{X}, \vec{\mathsf{V}}) = 0] \\
&= \frac{1}{2}\Pr[\mathsf{ab}_i] \\
&\quad + \sum_{\vec{\mathsf{v}}} \times Pr[X_i \wedge \mathsf{flg}_i(\vec{X}, \vec{\mathsf{V}}) = 0 \wedge \vec{\mathsf{V}} = \vec{\mathsf{v}}] \qquad (30) \\
&= \frac{1}{2}\Pr[\mathsf{ab}_i] + \sum_{\vec{\mathsf{v}}} \times Pr[\mathsf{flg}_i(\vec{X}, \vec{\mathsf{V}}) = 0 | X_i \wedge \vec{\mathsf{V}} = \vec{\mathsf{v}}] \\
&\quad \times \Pr[X_i \wedge \vec{\mathsf{V}} = \vec{\mathsf{v}}] \qquad (31) \\
&= \frac{1}{2}\Pr[\mathsf{ab}_i] + \sum_{\vec{\mathsf{v}}} \times Pr[\mathsf{flg}_i(\vec{X}, \vec{\mathsf{V}}) = 0 | \vec{\mathsf{V}} = \vec{\mathsf{v}}] \\
&\quad \times \Pr[X_i \wedge \vec{\mathsf{V}} = \vec{\mathsf{v}}] \qquad (32) \\
&= \frac{1}{2}\Pr[\mathsf{ab}_i] + \sum_{\vec{\mathsf{v}}} \times Pr[\mathsf{flg}_i(\vec{X}, \vec{\mathsf{v}}) = 0] \\
&\quad \times \Pr[X_i \wedge \vec{\mathsf{V}} = \vec{\mathsf{v}}] \qquad (33) \\
&= \frac{1}{2}\Pr[\mathsf{ab}_i] + \sum_{\vec{\mathsf{v}}} \lambda(\vec{\mathsf{v}}) \Pr[X_i \wedge \vec{\mathsf{V}} = \vec{\mathsf{v}}]. \qquad (34)
\end{aligned}$$

Steps (30) to (34) are based on a similar calculation in Bellare and Ristenpart (2009).

$$\begin{aligned}
&\Pr[Y_1] - \Pr[Y_2] \\
&= \sum_{\vec{v}} \lambda(\vec{v})(\Pr[X_1 \wedge \vec{V} = \vec{v}] - \Pr[X_2 \wedge \vec{V} = \vec{v}]).
\end{aligned}$$

Using $\lambda^- \leq \lambda(\vec{v}) \leq \lambda^+$ and $\sum_{\vec{v}} \Pr[X_i \wedge \vec{V} = \vec{v}] = \Pr[X_i]$, we obtain

$$\begin{aligned}
\lambda^- \Pr[X_1] - \lambda^+ \Pr[X_2] &\leq \Pr[Y_1] - \Pr[Y_2] \\
&\leq \lambda^+ \Pr[X_1] - \lambda^- \Pr[X_2].
\end{aligned}$$

Dividing throughout by λ^+ and using $\lambda^-/\lambda^+ = (1 - q/m)$ from Proposition 1, we get,

$$\begin{aligned}
&\Pr[X_1] - \Pr[X_2] - \frac{q}{m}\Pr[X_1] \\
&\leq \frac{\Pr[Y_1] - \Pr[Y_2]}{\lambda^+} \\
&\leq \Pr[X_1] - \Pr[X_2] + \frac{q}{m}\Pr[X_2]
\end{aligned}$$

This shows

$$\Pr[X_1] - \Pr[X_2] - \frac{\Pr[Y_1] - \Pr[Y_2]}{\lambda^+} \leq \frac{q}{m}\Pr[X_1] \leq \frac{q}{m}$$

and

$$\Pr[X_1] - \Pr[X_2] - \frac{\Pr[Y_1] - \Pr[Y_2]}{\lambda^+} \geq -\frac{q}{m}\Pr[X_2] \geq -\frac{q}{m}.$$

Combining these two bounds we obtain

$$-\frac{q}{m} \leq (\Pr[X_1] - \Pr[X_2]) - \frac{\Pr[Y_1] - \Pr[Y_2]}{\lambda^+} \leq \frac{q}{m}.$$

As a result,

$$\begin{aligned}
&|\Pr[X_1] - \Pr[X_2]| - \left|\frac{\Pr[Y_1] - \Pr[Y_2]}{\lambda^+}\right| \\
&\leq \left|(\Pr[X_1] - \Pr[X_2]) - \frac{\Pr[Y_1] - \Pr[Y_2]}{\lambda^+}\right| \leq \frac{q}{m}.
\end{aligned}$$

Assuming (ε_{dbdh}, t') hardness of DBDH, $|\Pr[Y_1] - \Pr[Y_2]| \leq \varepsilon_{dbdh}$. Using this, we get

$$|\Pr[X_1] - \Pr[X_2]| \leq \left|\frac{\Pr[Y_1] - \Pr[Y_2]}{\lambda^+}\right| + \frac{q}{m} \\ \leq \frac{\varepsilon_{dbdh}}{\lambda^+} + \frac{q}{m}.$$

The proof of Proposition 7.6 is now completed as follows.

$$\begin{aligned}\varepsilon_{hibe} &= \left|\Pr[X_0] - \frac{1}{2}\right| \\ &= |\Pr[X_0] - \Pr[X_2]| \\ &\leq |\Pr[X_0] - \Pr[X_1]| + |\Pr[X_1] - \Pr[X_2]| \\ &\leq \frac{\varepsilon_{dbdh}}{\lambda^+} + \frac{q}{m}.\end{aligned}$$

□

Since m was chosen to be at least $2q/\varepsilon_{hibe}$, the proof of Theorem 2 follows by substituting the value of m in the statement of Proposition 4. □

4.5 Bounds on probability of not abort

We require the following independence results in obtaining the required bound on the probability of abort. Similar independence results have been used in Waters (2005); Naccache (2007) in connection with IBE schemes.

Proposition 5. Let $L(\cdot)$ be as defined in (10). Let $\mathsf{v}_1, \ldots, \mathsf{v}_j$ be identities, i.e., each $\mathsf{v}_i = (\mathsf{v}_{i,1}, \ldots, \mathsf{v}_{i,l})$, with $\mathsf{v}_{i,k}$ to be an n/l-bit string (and hence $0 \leq \mathsf{v}_{i,k} \leq 2^{n/l} - 1$) and let $\theta \in \{1, \ldots, j\}$.

1 $\quad \mathsf{Pr}\left[L_\theta(\mathsf{v}_\theta) = 0 \,\middle|\, \bigwedge_{k=1, k\neq\theta}^{j} (L_k(\mathsf{v}_k) = 0)\right] = \dfrac{1}{m}.$

2 $\quad$ Let v'_θ be an identity such that $\mathsf{v}'_\theta \neq \mathsf{v}_\theta$. Then $\mathsf{Pr}\left[L_\theta(\mathsf{v}'_\theta) = 0 \,\middle|\, \bigwedge_{k=1}^{j} (L_k(\mathsf{v}_k) = 0)\right] = \dfrac{1}{m}.$

The probability is over the independent and uniform random choices of $x'_1, \ldots, x'_j, x_1, \ldots, x_l$ from $\mathbb{Z}_m$.

Proof. Recall from (10) that $L_k(\mathsf{v}_k) = x'_k + \mathsf{v}_{k,1}x_1 + \cdots + \mathsf{v}_{k,l}x_l$. The values $x'_1, \ldots, x'_j, x_1, \ldots, x_l$ are chosen independently and uniformly at random from $\mathbb{Z}_m$. In particular, the independence of $x'_1, \ldots, x'_j$ ensure that $L_1(\mathsf{v}_1), \ldots, L_j(\mathsf{v}_j)$ are also independently and uniformly distributed over $\mathbb{Z}_m$. The first point follows from this observation.

For the second point, without loss of generality, we may assume that $\theta = j$, since otherwise we may rename variables to achieve this. Since $\mathsf{v}_j \neq \mathsf{v}'_j$, there is an $i \in \{1, \ldots, l\}$ such that not both of $\mathsf{v}_{j,i}$ and $\mathsf{v}'_{j,i}$ are zeros. Without loss of generality, suppose that $\mathsf{v}'_{j,i}$ is non-zero. Then the result follows from the independent and uniform randomness of $x'_1, \ldots, x'_j, x_i$. □

Proof of Proposition 1: For any fixed $\vec{\mathsf{v}}$, let $\mathsf{ab}(\vec{\mathsf{v}})$ be the event $\mathsf{flg}(\vec{X}, \vec{\mathsf{v}}) = 1$. For $1 \leq i \leq q$, let E_i denote the event that the simulator does not abort on the i^{th} key extraction query and let C be the event that the simulator does not abort in the challenge stage.

We first consider the event C. Suppose the challenge identity is $\mathsf{v}^* = (\mathsf{v}^*_1, \ldots, \mathsf{v}^*_{h^*})$. Event C holds if and only if $F_j(\mathsf{v}^*_j) \equiv 0 \bmod p$ for $1 \leq j \leq h^*$. Recall that by choice of p, we can assume $F_j(\mathsf{v}^*_j) \equiv 0 \bmod p$ if and only if $x'_j + \sum_{k=1}^{l} x_k \mathsf{v}_{j,k} = mk_j$. Hence,

$$\mathsf{Pr}[C] = \mathsf{Pr}\left[\bigwedge_{j=1}^{h^*}\left(x'_j + \sum_{k=1}^{l} x_k \mathsf{v}_{j,k} = mk_j\right)\right]. \tag{35}$$

For $1 \leq j \leq h^*$ and $0 \leq i \leq \mu_l$, denote the event $x'_j + \sum_{k=1}^{l} x_k \mathsf{v}_{j,k} = mi$ by $A_{j,i}$ and the event $k_j = i$ by $B_{j,i}$. Also, let $C_{j,i}$ be the event $A_{j,i} \wedge B_{j,i}$.

Note that the event $\bigvee_{i=0}^{\mu_l} A_{j,i}$ is equivalent to the condition $x'_j + \sum_{k=1}^{l} x_k \mathsf{v}_{j,k} \equiv 0 \bmod m$ and hence equivalent to the condition $L_j(\mathsf{v}_j) = 0$. Since k_j is chosen uniformly at random from the set $\{0, \ldots, \mu_l\}$, we have $\mathsf{Pr}[B_{j,i}] = 1/(1 + \mu_l)$ for all j and i. The events $B_{j,i}$'s are independent of each other and also independent of the $A_{j,i}$'s. We have

$$\begin{aligned}
&\mathsf{Pr}\left[\bigwedge_{j=1}^{h^*}\left(x'_j + \sum_{k=1}^{l} x_k \mathsf{v}_{j,k} = mk_j\right)\right] \\
&= \mathsf{Pr}\left[\bigwedge_{j=1}^{h^*}\left(\bigvee_{i=0}^{\mu_l} C_{j,i}\right)\right] \\
&= \mathsf{Pr}\left[\bigvee_{i_1,\ldots,i_{h^*} \in \{0,\ldots,\mu_l\}} (C_{1,i_1} \wedge \cdots \wedge C_{h^*,i_{h^*}})\right] \\
&= \mathsf{Pr}\Big[\bigvee_{i_1,\ldots,i_{h^*} \in \{0,\ldots,\mu_l\}} \\
&\qquad (A_{1,i_1} \wedge B_{1,i_1} \wedge \cdots \wedge A_{h^*,i_{h^*}} \wedge B_{h^*,i_{h^*}})\Big] \\
&= \textstyle\sum_{i_1,\ldots,i_{h^*} \in \{0,\ldots,\mu_l\}} \\
&\qquad \mathsf{Pr}\left[A_{1,i_1} \wedge B_{1,i_1} \wedge \cdots \wedge A_{h^*,i_{h^*}} \wedge B_{h^*,i_{h^*}}\right] \\
&= \textstyle\sum_{i_1,\ldots,i_{h^*} \in \{0,\ldots,\mu_l\}} (\mathsf{Pr}\left[A_{1,i_1} \wedge \cdots \wedge A_{h^*,i_{h^*}}\right] \\
&\qquad\qquad \times \mathsf{Pr}\left[B_{1,i_1} \wedge \cdots \wedge B_{h^*,i_{h^*}}\right]) \\
&= \tfrac{1}{(1+\mu_l)^{h^*}} \textstyle\sum_{i_1,\ldots,i_{h^*} \in \{0,\ldots,\mu_l\}} \mathsf{Pr}\left[A_{1,i_1} \wedge \cdots \wedge A_{h^*,i_{h^*}}\right] \\
&= \tfrac{1}{(1+\mu_l)^{h^*}} \mathsf{Pr}\left[\bigvee_{i_1,\ldots,i_{h^*} \in \{0,\ldots,\mu_l\}} (A_{1,i_1} \wedge \cdots \wedge A_{h^*,i_{h^*}})\right] \\
&= \tfrac{1}{(1+\mu_l)^{h^*}} \mathsf{Pr}\left[\bigwedge_{j=1}^{h^*}\left(\bigvee_{i=0}^{\mu_l} A_{j,i}\right)\right] \\
&= \tfrac{1}{(1+\mu_l)^{h^*}} \mathsf{Pr}\left[\bigwedge_{j=1}^{h^*} (L_j(\mathsf{v}_j) = 0)\right] \\
&= \tfrac{1}{(m(1+\mu_l))^{h^*}}.
\end{aligned}$$

The last equality follows from Proposition 5(1). This shows that $\mathsf{Pr}[C] \leq 1/(m(1 + \mu_l)^h)$ and so

$$\mathsf{Pr}[\overline{\mathsf{ab}(\vec{\mathsf{v}})}] = \mathsf{Pr}\left[\left(\bigwedge_{i=1}^{q} E_i\right) \wedge C\right] \\ \leq \mathsf{Pr}[C] \leq \frac{1}{(m(1+\mu_l))^{h^*}}.$$

Since $1 \le h^* \le h$, we have $\Pr[C] \le \min_{1\le i\le h} 1/(m(1+\mu_l))^{h^*} = 1/(m(1+\mu_l))^h$. This shows the required upper bound.

For the lower bound, first note the following calculation.

$$\begin{aligned}
\Pr[\overline{\mathsf{ab}(\vec{\mathsf{v}})}] &= \Pr\left[\left(\bigwedge_{i=1}^{q} E_i\right) \wedge C\right] \\
&= \Pr\left[\left(\bigwedge_{i=1}^{q} E_i\right) |C\right] \Pr[C] \\
&= \left(1 - \Pr\left[\left(\bigvee_{i=1}^{q} \neg E_i\right) |C\right]\right) \Pr[C] \\
&\ge \left(1 - \sum_{i=1}^{q} \Pr\left[\neg E_i\, |C\right]\right) \Pr[C].
\end{aligned}$$

We now turn to bounding $\Pr[\neg E_i|C]$. For simplicity of notation, we will drop the subscript i from E_i and consider the event E that the simulator does not abort on a particular key extraction query on an identity $(\mathsf{v}_1, \ldots, \mathsf{v}_j)$. By the simulation, the event $\neg E$ implies that $L_i(\mathsf{v}_i) = 0$ for all $1 \le i \le j$. This holds even when the event is conditioned under C. Thus, we have $\Pr[\neg E|C] \le \Pr[\wedge_{i=1}^{j} L_i(\mathsf{v}_i) = 0|C]$. The number of components in the challenge identity is h^* and now two cases can happen:

$j \le h^*$: By the scheme constraint (a prefix of the challenge identity cannot be queried to the key extraction oracle), we must have a θ with $1 \le \theta \le j$ such that $\mathsf{v}_\theta \ne \mathsf{v}^*_\theta$.

$j > h^*$: In this case, we choose $\theta = h^* + 1$.

Now we have

$$\begin{aligned}
\Pr[\neg E|C] &\le \Pr\left[\bigwedge_{i=1}^{j} L_i(\mathsf{v}_i) = 0|C\right] \\
&\le \Pr[L_\theta(\mathsf{v}_\theta) = 0|C] \\
&= \Pr\left[L_\theta(\mathsf{v}_\theta) = 0|\bigwedge_{i=1}^{h^*} L_i(\mathsf{v}^*_i) = 0\right] = 1/m.
\end{aligned}$$

The last equality follows from an application of either Proposition 5(1) or Proposition 5(2) according as whether $j > h^*$ or $j \le h^*$. Substituting this in the bound for $\Pr[\overline{\mathsf{ab}(\vec{\mathsf{v}})}]$ we obtain

$$\begin{aligned}
\Pr[\overline{\mathsf{ab}(\vec{\mathsf{v}})}] &\ge \left(1 - \sum_{i=1}^{q} \Pr\left[\neg E_i\, |C\right]\right) \Pr[C]. \\
&\ge \left(1 - \frac{q}{m}\right) \frac{1}{(m(\mu_l + 1))^{h^*}} \\
&\ge \left(1 - \frac{q}{m}\right) \frac{1}{(m(\mu_l + 1))^{h}}.
\end{aligned}$$

This completes the proof of Proposition 1. □

4.6 Comparison of the two analyses

The two analyses given above lead to two different bounds. Since the artificial abort step requires $\mathcal{B}$ to possibly abort even when the game has been successfully completed, one would expect that doing away with this step should give a better bound. Perhaps somewhat counter-intuitively, the second analysis, which does not require artificial abort leads to a worse bound compared to the first analysis. We provide an explanation for this apparent anomaly.

The first thing to note is that the essential aim of both the analyses is to ensure that the abort (usual and also possibly artificial) takes place with more or less the same probability irrespective of the adversarial transcript. In the analysis for Theorem 1, the event $\overline{\mathsf{ab}}$ consists of both usual and artificial abort. From (29), we have

$$\lambda - \frac{\lambda\epsilon}{2} \le \Pr[\overline{\mathsf{ab}}|\gamma = \gamma'], \Pr[\overline{\mathsf{ab}}|\gamma \ne \gamma'] \le \lambda + \frac{\lambda\epsilon}{2}.$$

In other words, this shows that the conditional probabilities of not aborting is close to λ which is a lower bound on the probability of not aborting during the simulation. By the choice of $m = 2q$ in Theorem 1, the value of λ has been worked out to be $\lambda = 1/(2(m(\mu_l + 1))^h) = 1/(2(2q(\mu_l + 1))^h)$.

On the other hand, the analysis of Theorem 2 does not have any aritificial abort step and directly uses Proposition 1 to bound the probability of not abort. This probability is between λ^- and λ^+ with $\lambda^- = \left(1 - \frac{q}{m}\right)\lambda^+$ and $\lambda^+ = \frac{1}{(m(\mu_l+1))^h}$. To ensure that λ^- and λ^+ are close, one requires m to be equal to $2q/\epsilon_{hibe}$. But, this then results in λ^+ taking the value $\frac{\epsilon^h_{hibe}}{(2q(\mu_l+1))^h}$. Comparing this to the probability of not aborting in the case of Theorem 1, one sees that in absolute terms, this probability is significantly lower. So, even though there is no artificial abort in the analysis of Theorem 2, due to the choice of parameters, the actual probability of not abort is lower than the corresponding probability for Theorem 1. Unsurprisingly enough, the security bound for Theorem 2 is worse than the bound for Theorem 1.

5 A hierarchical identity-based signature scheme

It is an observation of Naor (as mentioned in Boneh and Franklin, 2003) that any IBE scheme can be converted into a signature scheme. The signer plays the role of the PKG, the message to be signed is treated as an identity and a signature is a decryption key for the identity. Verification of a signature on a message consists of encrypting a random string using the message as the identity and then determining whether decryption using the given signature (i.e., the decryption key for the identity) gives back the string. If the IBE is CPA-secure, then the constructed signature scheme is secure in the sense of existential unforgeability under a chosen-message attack.

The above idea has been used to construct several signature schemes. For example, Boneh et al. (2001) scheme is constructed from the IBE scheme by Boneh and Franklin (2003) while the IBE scheme of Gentry (2006) on hind sight gives the signature scheme of Boneh and Boyen (2004). Waters (2005) himself had constructed a signature scheme based on his IBE scheme and our conference paper had described a signature scheme obtained from the generalisation of Waters' (2005) IBE scheme.

The method of converting an IBE to a signature scheme extends to convert a HIBE scheme into a hierarchical identity-based signature (HIBS) scheme. This has been noted in Gentry and Silverberg (2002). An entity with identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$ has a corresponding decryption key d_{v}. Signature on a message M is then simply the decryption key for the identity tuple $(\mathsf{v}_1, \ldots, \mathsf{v}_j, M)$ which can be generated using d_{v}, which now acts as a signing key. So an $(h+1)$-level HIBE scheme gives rise to an h-level HIBS scheme. Though conceptually simple, there is a problematic issue. For such a scheme, the message space and the identity space should be disjoint. As otherwise, someone possessing the signature of M for the identity $(\mathsf{v}_1, \ldots, \mathsf{v}_j)$ can now use this signature as a signing key to produce a forgery for a message M' under the identity $(\mathsf{v}_1, \ldots, \mathsf{v}_j, M)$.

Paterson and Schuldt (2006) had described an identity-based signature (IBS) scheme based on 2-level Waters (2005) HIBE. The scheme has been proved to be secure. Now, consider the situation where the individual entities *and* the PKG *both* sign messages. The PKG signs messages using the master secret key and an individual entity signs messages using the signing key obtained from the PKG. For such a scenario, the scheme in Paterson and Schuldt (2006) is no longer secure. Basically, the above situation applies. An adversary may obtain the PKG's signature on a 'message' M and then use this signature as the signing key to sign messages under the 'identity' M. This happens due to the fact that the scheme in Paterson and Schuldt (2006) does not ensure that the message and identity spaces are disjoint. Gentry and Silverberg noted this issue of separating the message space and the identity space while describing their HIBS scheme in Gentry and Silverberg (2002) and suggested the use of a bit-prefix to separate the two spaces. We also use a similar technique to avoid this problem.

Below we describe the construction of two HIBS scheme. The first scheme, HIBS-1 is obtained from the HIBE scheme described in Section 3.1. A signature scheme is obtained as a particular case. A new IBS is also obtained as a particular case. Since we ensure that the message and identity spaces are (computationally) disjoint, individual users as well as the PKG can securely sign messages. The major improvement of the new IBS scheme upon the IBS scheme in Paterson and Schuldt (2006) is that the size of the public parameters is reduced by almost half. This improvement is a result of the reduction in public parameters of the HIBE in Section 3.1 over the HIBE in Waters (2005).

The second HIBS scheme is motivated by the goal of reducing the length of the signature. We show that one can achieve this at the cost of increasing the size of the public parameters. A similar trade-off between the sizes of the ciphertext and public parameters is obtained in the construction of HIBE-1 versus HIBE-2. But, the HIBS-2 scheme that we define below is not obtained from HIBE-2. In HIBE-2, the decryption key consists of elements of $\mathbb{G}_2$. In the signature setting this decryption key would correspond to a signature and so a direct conversion of HIBE-2 to a HIBS scheme would result in the signature consisting of elements of $\mathbb{G}_2$. Instead, we describe a variant, where the signature components are elements of $\mathbb{G}_1$, thus reducing the size of the signature. As one may surmise, this suggests that there is a corresponding HIBE where decryption keys consist of elements of $\mathbb{G}_1$ and the ciphertext consists of elements of $\mathbb{G}_2$. In the HIBE setting, this seems to be of less practical importance and so we did not provide the details of this construction.

5.1 HIBS

A HIBS scheme consists of four algorithms (which are probabilistic and polynomial time in the security parameter): Set-Up, KeyGen, Sign and Verify. For a HIBS of height h (henceforth denoted as h-HIBS) any identity v is a tuple $(\mathsf{v}_1, \ldots, \mathsf{v}_j)$ where $1 \leq j \leq h$.

- $\mathsf{HIBS.SetUp}$ and $\mathsf{HIBS.KeyGen}(\mathsf{v}, d_{\mathsf{v}|_{j-1}}, pk)$ are exactly the same as that of a HIBE scheme.
- $\mathsf{HIBS.Sign}(\mathsf{v}, d_{\mathsf{v}}, M, pk)$. Takes as input v, a decryption key d_{v} for v, the message M and pk, and returns sig, the signature of M under the identity v.
- $\mathsf{HIBS.Verify}(\mathsf{v}, M, \mathsf{sig}, pk)$. Takes as input v, message M, signature sig and outputs yes if sig is a valid signature for M under v or if this does not hold, then it outputs no.

The security model for existential unforgeability under chosen message attacks consists of a game between an adversary and a simulator and goes through the following phases.

Set-up. The simulator sets up the HIBS scheme, i.e., generates the public parameter pk and the master secret key for the scheme and provides the adversary with pk.

Queries. The adversary makes two types of queries in an interleaved and adaptive manner.

- Extract queries. The adversary can ask for the private key of any identity. The simulator provides a private key for this identity and the distribution of the private key should be the same as that generated by HIBS.KeyGen.
- Signature queries. In this type of query, the adversary provides an identity and a message. The simulator has to provide a proper signature on the message under the given identity.

Forgery. At the end of the interaction, the adversary outputs a message M^*, an identity v^* and a signature sig^*. The adversary is successful if the followings hold.

- $\mathsf{HIBS.Verify}(\mathsf{v}^*, M^*, \mathsf{sig}^*, pk)$ returns yes.
- The adversary has not made any previous key extraction query on v^* or any of its prefixes.
- The adversary has not made any previous signature query on (M^*, v^*).

The advantage of an adversary is defined to be the probability that the adversary succeeds in the above game. As usual, the HIBS scheme is said to be $(t, q_{ID}, q_S, \epsilon)$-secure if the advantage of any adversary which runs in time t, makes q_{ID} key extraction queries and q_S signature queries is at most ϵ.

Note that by the third condition of the above definition, the adversary cannot win by forging a 'new' signature on an 'old' message. An weaker alternative requirement to the third condition is to only require that the triplet $(M^*, \mathsf{v}^*, \mathsf{sig}^*)$ is 'new'. A signature scheme which is secure against such adversaries is said to be strongly unforgeable.

5.2 HIBS-1

Let $H : \{0,1\}^* \to \{0,1\}^n$ be a collision resistant hash function. The output of H is assumed to consist of l blocks where each block is an n/l-bit string considered to be an element of the set $\{0, \ldots, 2^{n/l} - 1\}$. Messages are assumed to be arbitrary binary strings and identities are assumed to be of the type $(\mathsf{str}_1, \ldots, \mathsf{str}_j)$, $1 \le j \le h$ and str_k is an arbitrary binary string. These are hashed into n-bit strings in the following manner. If msg is a message, then compute $H(0||\mathsf{msg})$, while if str_k is a component of an identity tuple then compute $H(1||\mathsf{str}_k)$. This ensures that n-bit strings obtained from messages will not be equal to n-bit strings obtained from identity components (assuming that H is collision resistant). Following the notation for our HIBE scheme, we will write $\mathsf{v}_k = H(1||\mathsf{str}_k)$.

With the above modifications, one can easily convert an $(h+1)$-level HIBE as decribed in Section 3.1 to an h-level HIBS. For an identity at the j^{th} level, the signature will be the private key of a $(j+1)$-level identity, with the message to be signed constituting the last level 'identity' in the hierarchy. The signature so constructed contains one element of $\mathbb{G}_1$ and $j+1$ elements of $\mathbb{G}_2$.

Set-up. The scheme is built from a Type 3 bilinear pairing setting $(p, \mathbb{G}_1 = \langle P_1 \rangle, \mathbb{G}_2 = \langle P_2 \rangle, \mathbb{G}_T, e)$. Suppose the maximum number of levels in the HIBS is h.

The PKG chooses random $U_1', \ldots, U_{h+1}', U_1, \ldots, U_l$ from $\mathbb{G}_1$. The PKG also chooses a random $R_1 \in \mathbb{G}_1$ and a random integer $\alpha \in \mathbb{Z}_p$ and computes $Q_2 = \alpha P_2$ and $e(R_1, Q_2)$. The public parameters are the following elements: P_2, $e(R_1, Q_2)$, $U_1', \ldots, U_{h+1}', U_1, \ldots, U_l$, The master secret is αR_1. The hash function H is also specified as part of the set-up.

Key generation. Let $(\mathsf{str}_1, \ldots, \mathsf{str}_j)$ be the identity for which a key has to be generated and let $\mathsf{v}_k = H(1||\mathsf{str}_k)$. A key corresponding to this identity is generated by essentially applying the key generation algorithm of the HIBE scheme in Section 3.1 to $(\mathsf{v}_1, \ldots, \mathsf{v}_j)$. For example, for a first level identity str_1, $\mathsf{v}_1 = H(1||\mathsf{str}_1)$ and the PKG computes the corresponding private key as $d_0 = \alpha R_1 + r_1 V_1^{(1)}(\mathsf{v}_1)$ and $d_1 = r_1 P_2$, where $V_1^{(1)}(\mathsf{v}_1) = U_1' + \sum_{i=1}^{l} \mathsf{v}_{1,i} U_i$ and r_1 is chosen randomly from $\mathbb{Z}_p^*$. Similarly, the entity with a signing key for $(\mathsf{v}_1, \ldots, \mathsf{v}_{j-1})$ (i.e., for $(\mathsf{str}_1, \ldots, \mathsf{str}_{j-1})$) can generate a signing key for $(\mathsf{v}_1, \ldots, \mathsf{v}_j)$ (i.e., for $(\mathsf{str}_1, \ldots, \mathsf{str}_j)$).

Signing. Suppose a message msg is to be signed under an identity $(\mathsf{str}_1, \ldots, \mathsf{str}_j)$. Let $\mathsf{v}_k = H(1||\mathsf{str}_k)$, for $1 \le k \le j$ and $\mathsf{v}_{j+1} = H(0||\mathsf{msg})$ and let $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$. Suppose that d_{v} is a signing key for v (i.e., for $(\mathsf{str}_1, \ldots, \mathsf{str}_j)$).

Then a signature on msg under the identity $(\mathsf{str}_1, \ldots, \mathsf{str}_j)$ is obtained by applying the key generation algorithm described above in the following manner. Using the key d_{v} for v, a key for the 'identity' $(\mathsf{v}_1, \ldots, \mathsf{v}_j, \mathsf{v}_{j+1})$ is created and this key is returned as the signature sig.

Verification. The input is a tuple $(\mathsf{msg}, (\mathsf{str}_1, \ldots, \mathsf{str}_j), \mathsf{sig})$ where sig is of the form $(d_0, d_1, \ldots, d_{j+1}) \in \mathbb{G}_1 \times \mathbb{G}_2^{j+1}$. Let $\mathsf{v}_k = H(1||\mathsf{str}_k)$, $1 \le k \le j$ and $\mathsf{v}_{j+1} = H(0||\mathsf{msg})$ and let $V_k = V_k^{(1)}(\mathsf{v}_k)$ for $1 \le k \le j+1$. The input is accepted if the following equality holds:

$$e(d_0, P_2) = e(R_1, Q_2) \times \prod_{k=1}^{j+1} e(V_k, d_i).$$

Correctness of the verification is similar to the correctness of the decryption for the HIBE-1 scheme in Section 3.1.

5.2.1 Comparison to previous signature schemes

The HIBS scheme described above has h levels and can be instantiated to obtain a usual signature scheme and an IBS. If we put $h = 0$, then we obtain a usual signature scheme where as if we put $h = 1$, then we obtain an IBS. For $h = 0$, the PKG is the signer as in Naor's transformation. For $h = 1$, the individual entities as well as the PKG can securely sign messages.

When considered as an IBS, the above scheme offers substantial reduction in the size of the public parameters as compared to the IBS scheme in Paterson and Schuldt (2006). While the IBS scheme in Paterson and Schuldt (2006) is described in the setting of symmetric pairing and requires $((2(l+1), 1, 1))$ size public parameters, the scheme described here is in the setting of asymmetric pairing and requires $(l+2, 1, 1, 0)$ size public parameters. There are two aspects to the efficiency improvement – the first one is due to working with asymmetric pairings and the second one is inherited from the efficiency improvement of the HIBE scheme in Section 3.1 over the HIBE scheme suggested by Waters (2005). While the scheme in Paterson and Schuldt (2006) can also be converted to asymmetric pairing setting, since it is based on the HIBE in Waters (2005), the size of public parameters will still be about double that of the scheme proposed here.

5.3 HIBS-2

We now suggest a second HIBS scheme where the signature consists of only elements of $\mathbb{G}_1$. This might be useful for certain applications where it is important to reduce the overall signature length.

$H : \{0,1\}^* \to \{0,1\}^n$ is a collision resistant hash function as in the construction of HIBS-1. The processing of messages and identities are the same as in HIBS-1 and we follow the same notation as in the case of HIBS-1.

Set-up. The scheme is built from a Type 3 bilinear pairing setting $(\mathbb{G}_1 = \langle P_1 \rangle, \mathbb{G}_2 = \langle P_2 \rangle, \mathbb{G}_T, e)$. Suppose the maximum number of levels in the HIBS is h. The PKG chooses random $x_i, y_j \in \mathbb{Z}_p^*$, where $1 \le i \le h+1$ and $1 \le j \le l$ and computes $U'_i = x_i P_1$, $W'_i = x_i P_2$, $U_j = y_j P_1$, $W_j = y_j P_2$. The PKG also chooses a random $R_1 \in \mathbb{G}_1$ and a random integer $\alpha \in \mathbb{Z}_p^*$ and computes $Q_2 = \alpha P_2$ and $e(R_1, Q_2)$. The public parameters are the following elements: P_2, $e(R_1, Q_2)$, $\overrightarrow{U} = (U'_1, \ldots, U'_{h+1}, U_1, \ldots, U_l)$, $\overrightarrow{W} = (W'_1, \ldots, W'_{h+1}, W_1, \ldots, W_l)$. The master secret is αR_1. The hash function H is also specified as part of the set-up.

$\overrightarrow{U} \in \mathbb{G}_1^{h+1+l}$ in the public parameter will be used by the signers (for both key generation and signing) to hash an identity or a message into an element of $\mathbb{G}_1$ and we will use the notation $V_{1,k}$ for this mapping. On the other hand, $\overrightarrow{W} \in \mathbb{G}_2^{h+1+l}$ will be used by the verifier to hash an identity or a message into an element of $\mathbb{G}_2$ and we will use the notation $V_{2,k}$ for this mapping. Note that, $V_{1,k}$ and $V_{2,k}$ are as defined by (2) and (3) and beacuse of the choice of $\overrightarrow{U}$ and $\overrightarrow{W}$, we have $\log_{P_1} V_{k,1} = \log_{P_2} V_{k,2}$.

Key generation. Let $(\mathsf{str}_1, \ldots, \mathsf{str}_j)$ be the identity for which a key has to be generated and let $\mathsf{v}_k = H(1||\mathsf{str}_k)$. A key corresponding to this identity is generated by essentially applying the key generation algorithm of the HIBE scheme in Section 3.1 to $(\mathsf{v}_1, \ldots, \mathsf{v}_j)$. For example, for a first level identity str_1, $\mathsf{v}_1 = H(1||\mathsf{str}_1)$ and the PKG computes the corresponding private key as $d_0 = \alpha R_1 + r_1 V_{1,1}(\mathsf{v}_1)$ and $d_1 = r_1 P_1$, where $V_{1,1}(\mathsf{v}_1) = U'_1 + \sum_{i=1}^{l} \mathsf{v}_{1,i} U_i$ and r_1 is chosen randomly from $\mathbb{Z}_p^*$. Similarly, the entity with a signing key for $(\mathsf{v}_1, \ldots, \mathsf{v}_{j-1})$ (i.e., for $(\mathsf{str}_1, \ldots, \mathsf{str}_{j-1})$) can generate a signing key for $(\mathsf{v}_1, \ldots, \mathsf{v}_j)$ (i.e., for $(\mathsf{str}_1, \ldots, \mathsf{str}_j)$). Note that the signing key thus generated contains elements of $\mathbb{G}_1$ only.

Signing. Suppose a message msg is to be signed under an identity $(\mathsf{str}_1, \ldots, \mathsf{str}_j)$. Let $\mathsf{v}_k = H(1||\mathsf{str}_k)$, for $1 \le k \le j$ and $\mathsf{v}_{j+1} = H(0||\mathsf{msg})$ and let $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$. Suppose that d_{v} is a signing key for v (i.e., for $(\mathsf{str}_1, \ldots, \mathsf{str}_j)$).

Then a signature on msg under the identity $(\mathsf{str}_1, \ldots, \mathsf{str}_j)$ is obtained by applying the key generation algorithm described above in the following manner. Using the key d_{v} for v, a key for the 'identity' $(\mathsf{v}_1, \ldots, \mathsf{v}_j, \mathsf{v}_{j+1})$ is created and this key is returned as the signature sig.

Verification. The input is a tuple $(\mathsf{msg}, (\mathsf{str}_1, \ldots, \mathsf{str}_j), \mathsf{sig})$ where sig is of the form $(d_0, d_1, \ldots, d_{j+1}) \in \mathbb{G}_1^{j+2}$.

Let $\mathsf{v}_k = H(1||\mathsf{str}_k)$, $1 \le k \le j$ and $\mathsf{v}_{j+1} = H(0||\mathsf{msg})$ and let $V_{2,i} = V_{2,i}(\mathsf{v}_i)$ for $1 \le i \le j+1$. The input is accepted if the following equality holds:

$$e(d_0, P_2) = e(R_1, Q_2) \times \prod_{i=1}^{j+1} e(d_i, V_{2,i}).$$

Correctness of the verification is similar to the correctness of the decryption for the HIBE scheme in Section 3.1. Trade-off between HIBS-1 and HIBS-2 is shown in Table 4.

5.4 Security

The security of HIBS-1 and HIBS-2 is based on the hardness of the co-CDH problem as defined in Section 2.4.

Theorem 3. The signature schemes HIBS-1 and HIBS-2 are $(t, q_{ID}, q_S, \epsilon_{hibs})$-secure in the sense of existential unforgeability under chosen message attacks under the assumption that the co-CDH problem in $(p, \mathbb{G}_1, \mathbb{G}_2, G_t, e)$ is $(\epsilon_{co\text{-}cdh}, t')$ hard with

$$\epsilon_{hibs} \le (2(m(\mu_l + 1))^{h+1})\epsilon_{co\text{-}cdh}$$

where $t' \le t + t_{sim}$ and t_{sim} is the simulation time, i.e., the time to generate q_{ID} private keys, sign q_S messages and verify one signature; $\mu_l = l(2^{n/l} - 1)$, $m = \max(2q, 2^{n/l})$ and $q = q_{ID} + q_S$. We further assume $m(1 + \mu_l) < p$.

Table 4 Trade-off between HIBS-1 and HIBS-2

Scheme	*Parameter sizes*			
	PP	*msk*	*pvt key*	*sig*
HIBS-1	$(h+l+2, 1, 1)$	$(1, 0, 0)$	$(1, j, 0)$	$(1, j+1, 0)$
HIBS-2	$(h+l+2, h+l+2, 1)$	$(1, 0, 0)$	$(j+1, 0, 0)$	$(j+2, 0, 0)$

Scheme	*Times*		
	key gen	*sign*	*ver*
HIBS-1	$1\left[\mathrm{H}_{n,l}^{(1)}\right]+1[\mathrm{SM}_1]+1[\mathrm{SM}_2]$	$1\left[\mathrm{H}_{n,l}^{(1)}\right]+1[\mathrm{SM}_1]+1[\mathrm{SM}_2]$	$(j+1)\left[\mathrm{H}_{n,l}^{(1)}\right]+[\mathrm{P}_{j+3}]$
HIBS-2	$1\left[\mathrm{H}_{n,l}^{(1)}\right]+2[\mathrm{SM}_1]$	$1\left[\mathrm{H}_{n,l}^{(1)}\right]+2[\mathrm{SM}_1]$	$(j+1)\left[\mathrm{H}_{n,l}^{(2)}\right]+[\mathrm{P}_{j+3}]$

Notes: Here j denotes the number of components in the identity for which the computations are done. Key generation time shows only the time for generating a key for the j^{th} level using a key for the $(j-1)^{\text{st}}$ level.

Note.

1. In the proof below, we will assume that the collision-resistant function H behaves like an injective function, i.e., for distinct inputs, the outputs of H are distinct. This allows a simplification of the theorem statement and makes the similarity to the security analysis for HIBE somewhat more clearer. Formally, it is possible to include the collision-resistant property of H in the theorem statement. This can be done following the approach in Rogaway (2006) for formalising the security of keyless hash functions. But, this would complicate the statement quite a bit. So, we have chosen the simpler approach.
2. The proof is similar to that of Theorem 1 and we only provide the main idea for the case of HIBS-2. The security of HIBS-1 can be established in an analogous manner.
3. A major difference from the proof of Theorem 1 is that the artificial abort step is not required in this case.

Proof. An instance of the co-CDH problem in $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ is a tuple (Q, zP_1, zP_2) where Q is a random element of $\mathbb{G}_1$. Basically, the simulator $\mathcal{B}$ sets up the HIBS scheme from the instance (Q, zP_1, zP_2) in a manner similar to that in the proof of Theorem 1. The parameter R_1 is set to be equal to Q; Q_2 is set to be equal to zP_2. We assume that the secret key is zR_1 which is also unknown to the simulator. (In fact, the solution to the co-CDH instance is to compute $zQ = zR_1$.) The parameters $U_1, \ldots, U_l$ and $U'_1, \ldots, U'_{h+1}$ are defined exactly as in the proof of Theorem 1 and can be computed from zP_1. In a similar fashion, the simulator can compute the parameters $W_1, \ldots, W_l$ and $W'_1, \ldots, W'_{h+1}$ from zP_2.

Private key queries and signature queries made by the adversary $\mathcal{A}$ are handled using essentially the same method for answering the private key queries in the proof of Theorem 1. This may result in the simulator $\mathcal{B}$ requiring to abort.

Finally, the adversary $\mathcal{A}$ outputs a forgery $(\mathsf{msg}, (\mathsf{str}_1, \ldots, \mathsf{str}_j), \mathsf{sig})$ for some j in $\{1, \ldots, h\}$. As before, let $\mathsf{v}_k = H(1||\mathsf{str}_k)$, for $1 \le k \le j$ and $\mathsf{v}_{j+1} = H(0||\mathsf{msg})$. Note that by the collision resistant assumption on H, v_{j+1} is not equal to the output of an application of H to any earlier identity component. Also, the adversary should not have made an earlier sign query on $(\mathsf{msg}, (\mathsf{str}_1, \ldots, \mathsf{str}_j))$.

The signature sig is of the form $(d_0 = zR_1 + \sum_{i=1}^{j+1} r_i V_{1,i}, d_1 = r_1 P_1, \ldots, d_{j+1} = r_{j+1} P_1)$. If any of the values $F_i(\mathsf{v}_i)$, $1 \le i \le j+1$ is non-zero, then $\mathcal{B}$ aborts (this is similar to the generation of challenge ciphertext in the proof of Theorem 1). Otherwise, $\mathcal{B}$ computes

$$\begin{aligned} r_i V_{1,i} = r_i V_{1,i}(\mathsf{v}_i) &= r_i (F_i(\mathsf{v}_i) zP_1 + J_i(\mathsf{v}_i) P_1) \\ &= r_i J_i(\mathsf{v}_i) P_1 \\ &= J_i(\mathsf{v}_i) d_i. \end{aligned}$$

Note that $\mathcal{B}$ can compute $J_i(\mathsf{v}_i)$ and since $d_i = r_i P_1$ is given, $\mathcal{B}$ can indeed compute $r_i V_i$.

The simulator $\mathcal{B}$ now computes zQ as follows.

$$\begin{aligned} zR_1 &= d_0 - \left(\sum_{i=1}^{j+1} r_i V_{1,i}\right) \\ &= zR_1 + \left(\sum_{i=1}^{j+1} r_i V_{1,i}\right) - \left(\sum_{i=1}^{j+1} r_i V_{1,i}\right) \\ &= zQ. \end{aligned}$$

Thus, contingent on the fact that the simulator $\mathcal{B}$ does not abort, we obtain an algorithm to solve the co-CDH problem. Further, we assume that if $\mathcal{B}$ has to abort, then it returns a random element of $\mathbb{G}_1$ as output which is equal to zQ with probability $1/p$. A lower bound on the probability that $\mathcal{B}$ does not abort is obtained from Proposition 1.

Let $\mathsf{succ}(\mathcal{B})$ be the event that $\mathcal{B}$ is successful and let $\mathsf{succ}(\mathcal{A})$ be the event that the forgery attempt of adversary $\mathcal{A}$ is successful. Also, let ab be the event that $\mathcal{B}$ aborts. Note that unlike Theorem 1, there is no artificial abort in this case.

$$\begin{aligned} \Pr[\mathsf{succ}(\mathcal{B})] &= \Pr[\mathsf{succ}(\mathcal{B})|\mathsf{ab}] \Pr[\mathsf{ab}] + \Pr[\mathsf{succ}(\mathcal{B})|\overline{\mathsf{ab}}] \Pr[\overline{\mathsf{ab}}] \\ &\ge \Pr[\mathsf{succ}(\mathcal{A})] \Pr[\overline{\mathsf{ab}}]. \end{aligned}$$

Using $\Pr[\mathsf{succ}(\mathcal{B})] \le \epsilon_{co\text{-}cdh}$, we have

$$\Pr[\mathsf{succ}(\mathcal{A})] \le \frac{\Pr[\mathsf{succ}(\mathcal{B})]}{\Pr[\overline{\mathsf{ab}}]} \le \frac{\epsilon_{co\text{-}cdh}}{\lambda^-}.$$

Since this relation holds for all adversaries $\mathcal{A}$ running in time t and making (q_{ID}, q_S) queries and substituting the value of λ^- from Proposition 1 with $m = 2q$, we obtain the required result. □

5.5 *A short signature scheme*

HIBS-2 when specialised to a signature scheme admits an optimisation which leads to a small size signature while avoiding the associated increase in the size of the public parameters. Also, since there are no identities, the role of the hash function H is limited to only mapping messages to elements of $\mathbb{Z}_p$; in particular, it is not required to use H to separate between message and identity spaces. In this section, we provide the details of this scheme. The security of the scheme follows from Theorem 3 by setting $h = 0$.

Set-up. The Type 3 bilinear pairing setting $(\mathbb{G}_1 = \langle P_1 \rangle, \mathbb{G}_2 = \langle P_2 \rangle, \mathbb{G}_T, e)$ is used. The signer chooses random $x, y_1, \ldots, y_l \in \mathbb{Z}_p^*$ and computes $W' = xP_2$, $W_j = y_j P_2$. The signer also chooses a random $R_1 \in \mathbb{G}_1$ and a random integer $\alpha \in \mathbb{Z}_p^*$ and computes $Q_2 = \alpha P_2$ and $e(R_1, Q_2)$. The public key consists of the following elements: $(P_2,\ e(R_1, Q_2),\ W, W_1, \ldots, W_l)$. The signing key is $(\alpha R_1, x, y_1, \ldots, y_l)$.

Table 5 CCA-secure HIBE: parameter sizes and costs of different operations

Parameter sizes				*Times*		
PP	*msk*	*pvt key*	*cpr txt*	*key gen*	*enc*	*dec*
$(h+l+1,1,1,0)$	$(1,0,0,0)$	$(2,j,0,0)$	$(j+1,1,0,0)$	$1\left[\mathrm{H}_{n,l}^{(1)}\right]$ $+1[\mathrm{SM}_1]+1[\mathrm{SM}_2]$	$j\left[\mathrm{H}_{n,l}^{(1)}\right]+1[\mathrm{E}]$ $+(j+2)[\mathrm{SM}_1]+1[\mathrm{SM}_2]$	$1[\mathrm{P}_2]+1[\mathrm{P}_{j+1}]$ $+1[\mathrm{SM}_1]$

Notes: The variable j refers to the number of components in the input identity tuple. Since R_1 is required only for decryption, we count it as part of the decryption key and not as part of the public parameters. Costs of symmetric key operations are not shown.

Signing. Suppose a message msg is to be signed and let $\mathsf{v}=H(\mathsf{msg})$. Choose a uniform random r from $\mathbb{Z}_p^*$ and define the signature to be (d_0,d_1), where $d_0=\alpha R_1+rV_{1,1}(\mathsf{v})$, $d_1=rP_1$, and $V_{1,1}(\mathsf{v})=\left(x+\sum_{j=1}^{l}\mathsf{v}_iy_j\right)P_1$. Note that the signature consists of two elements of $\mathbb{G}_1$.

Verification. The input is a tuple $(\mathsf{msg},\mathsf{sig})$ where sig is of the form (d_0,d_1). Let $\mathsf{v}=H(\mathsf{msg})$ and let $V_{2,1}=V_{2,1}(\mathsf{v})$. The input is accepted if the following equality holds:

$$\begin{aligned} e(R_1,Q_2) &= e(d_0,P_2)\times e(d_1,-V_{2,1}) \\ &= \frac{e(d_0,P_2)}{e(d_1,V_{2,1})}. \end{aligned}$$

Note Compared to HIBS-2 with $h=0$, the Us are not required and this saves $(l+1)$ elements of $\mathbb{G}_1$ from the public parameters. There is an increase in the size of the signing key by $(l+1)$ elements of $\mathbb{Z}_p$. Depending on the application, this may not be significant. For example, if the signing is to be done by a server, then the storage of the signing key is not an issue, but the transmission of the signatures will be important. On the other hand, if the signing is to be done using a smart card, then requiring a small signing key becomes more important and the trade-off obtained by this scheme is less relevant.

This provides the smallest size signature among all schemes whose security is based on a static assumption and whose security reduction does not assume any function to be a random oracle.

The signature scheme described by Waters (2005) based on his IBE scheme also has two group elements as the signature. But, the description is in the setting of Type 1 pairing (symmetric pairing) and so, the lengths of representations of these two group elements will be significantly longer than the lengths of representations of two elements of $\mathbb{G}_1$. A straightforward conversion of Waters signature to asymmetric pairing setting will give a scheme whose signatures consist of one element of $\mathbb{G}_1$ and one element of $\mathbb{G}_2$. Achieving a signature scheme in the Type 3 setting where the signature consists of two elements of $\mathbb{G}_1$ is not a completely routine task.

6 CCA-secure HIBE protocol

In this section, we modify the CPA-secure HIBE-1 scheme of Section 3.1 to obtain a CCA-secure HIBE scheme. We provide an explicit hybrid scheme. This allows us to improve the decryption efficiency as we explain later. The modification consists of certain additions to the set-up procedure as well as modifications of the encryption and the decryption algorithms. No changes are required in the key generation algorithm.

The HIBE-2 scheme of Section 3.2 can also be converted in a similar manner to yield a CCA-secure HIBE scheme. The details are very similar to the case of HIBE-1 and are omitted. All the trade-offs of HIBE-1 versus HIBE-2 also hold under conversion to CCA-secure schemes. When specialised to an IBE scheme, HIBE-2 admits an additional optimisation. This is discussed later and we also provide the details of the IBE version of HIBE-2 scheme.

The additions to HIBE-1 are based on the technique used by Boyen-Mei-Waters (Boneh et al., 2005b) and are also based on the IBE construction by Boneh and Boyen (2004a) (BB-IBE). The BMW and the BB techniques are described in the setting of symmetric pairing and we convert these into the setting of asymmetric pairing. Some new ideas – incorporating length of the identity into the ciphertext and using symmetric key authentication to verify ciphertext well formedness – are introduced. Also, an AE scheme is used to combine the two tasks of symmetric key encryption and authentication.

The description of the construction is given in Figure 1. The bold portions of Figure 1 provide the additional points required over the CPA-secure HIBE construction of Section 3.1. We provide some intuition of how decryption queries are answered. First, let us consider what happens if we attempt to simulate decryption queries by key extraction queries. The idea is that we use a key extraction query to derive the private key of the identity which is provided as part of the decryption query. Then this private key is used to decrypt the ciphertext. This idea works fine except for the situation where a decryption query is made on a prefix of the challenge identity. Since, it is not allowed to query the key extraction oracle on prefixes of the challenge identity, the above simulation technique will not work. We need an additional mechanism to answer such decryption queries.

Figure 1 CCA-secure HIBE

HIBE.SetUp

1. Choose α uniformly at random from $\mathbb{Z}_p$.
2. Set $Q_2 = \alpha P_2$.
3. Choose $R_1, U'_1, \ldots, U'_h, U_1, \ldots, U_l$ randomly from $\mathbb{G}_1$.
4. **Choose $\mathbf{W}$ randomly from $\mathbb{G}_1$.**
5. **Let $\mathbf{H_s} : \{\mathbf{1}, \ldots, \mathbf{h}\} \times \mathbb{G}_2 \to \mathbb{Z}_p$ be chosen from a UOWHF and made public.**
6. Public parameters:
   R_1 (required only for decryption),
   $P_2, e(R_1, Q_2), U'_1, \ldots, U'_h, U_1, \ldots, U_l$ and $\mathbf{W}$.
7. Master secret key: αR_1.

HIBE.KeyGen: Identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$.

1. Choose $r_1, \ldots, r_j$ randomly from $\mathbb{Z}_p$.
2. $d_0 = \alpha R_1 + \sum_{k=1}^{j} r_k V_k^{(1)}(\mathsf{v}_k)$.
3. $d_k = r_k P_2$ for $k = 1, \ldots, j$.
4. Output $d_{\mathsf{v}} = (d_0, d_1, \ldots, d_j)$.

HIBE.Encrypt: Identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$; message M.

1. Choose t randomly from $\mathbb{Z}_p$.
2. $C_1 = tP_2$, $B_1 = tV_1^{(1)}(\mathsf{v}_1), \ldots, B_j = tV_j^{(1)}(\mathsf{v}_j)$.
3. $K = e(R_1, Q_2)^t$.
4. $(\mathsf{IV}, dk) = \mathsf{KDF}(K)$.
5. $(\mathsf{cpr}, \mathsf{tag}) = \mathsf{AE.Encrypt}_{dk}(\mathsf{IV}, M)$.
6. $\upsilon = \mathbf{H_s}(\mathbf{j}, \mathbf{C_1})$; $\mathbf{W}_\upsilon = \mathbf{W} + \upsilon \mathbf{R_1}$; $\mathbf{C_2} = \mathbf{tW}_\upsilon$.
7. Output $(C_1, \mathbf{C_2}, B_1, \ldots, B_j, \mathsf{cpr}, \mathsf{tag})$.

HIBE.Decrypt: Identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$; ciphertext $(C_1, \mathbf{C_2}, B_1, \ldots, B_j, \mathsf{cpr}, \mathsf{tag})$; decryption key $d_{\mathsf{v}} = (d_0, d_1, \ldots, d_j)$.

1. $\upsilon = \mathbf{H_s}(\mathbf{j}, \mathbf{C_1})$; $\mathbf{W}_\upsilon = \mathbf{W} + \upsilon \mathbf{R_1}$.
2. **If $\mathbf{e}(\mathbf{W}_\upsilon, \mathbf{C_1}) \neq \mathbf{e}(\mathbf{C_2}, \mathbf{P_2})$ return $\perp$.**
3. $K = e(d_0, C_1) \times \prod_{k=1}^{j} e(B_k, -d_k)$.
4. $(\mathsf{IV}, dk) = \mathsf{KDF}(K)$.
5. $M = \mathsf{AE.Decrypt}_{dk}(\mathsf{IV}, C, \mathsf{tag})$.
   (This may abort and return $\perp$).
6. Output M.

Notes: 1 Maximum depth of the HIBE is h.
2 Identities are of the form $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$, $j \in \{1, \ldots, h\}$, $\mathsf{v}_k = (\mathsf{v}_{k,1}, \ldots, \mathsf{v}_{k,l})$ and $\mathsf{v}_{k,i}$ is an (n/l)-bit string.
3 The setting $(p, \mathbb{G}_1 = \langle P_1 \rangle, \mathbb{G}_2 = \langle P_2 \rangle, \mathbb{G}_T, e)$ is of Type 3 pairing as defined in Section 2.3.
4 The notation $V_k^{(1)}()$ is given in (2).
5 Key generation is the same as the scheme in Section 3.1.

The mechanism that we have used is primarily based on the BMW technique (Boneh et al., 2005b). The parameter W along with R_1 and Q_2 define an instance of a BB-IBE scheme. During encryption, an 'identity' $\upsilon = H_s(j, C_1)$ for this scheme is generated from the randomiser $C_1 = tP_2$ and the length j of the identity tuple. Using this identity, a separate encapsulation of the key $e(R_1, Q_2)^t$ is made. This encapsulation consists of the element C_2 (and C_1). In the security proof, if a decryption query is made on the challenge identity, then this encapsulation is used to obtain the private key of υ and answer the decryption query.

The use of the function $H_s()$ is different from its use in Boneh et al. (2005b). In Boneh et al. (2005b), the function $H_s()$ maps $\mathbb{G}_1$ to $\mathbb{Z}_p$. On the other hand, in the HIBE scheme in Figure 1, $H_s()$ maps $\{1, \ldots, h\} \times \mathbb{G}_2$ to $\mathbb{Z}_p$. Our aim is to include information about the length of the identity into the output of $H_s()$. Without this information, an encryption for a $(j+1)$-level identity can be converted to an encryption for its j-level prefix by simply dropping the term corresponding to the last component in the identity.

The other aspect is that of checking for the well formedness of the ciphertext. A well formed ciphertext requires verifying that $C_1 = tP_2$, $C_2 = tW_\upsilon$ and $B_1 = tV_1(\mathsf{v}_1), \ldots, B_j = tV_j(\mathsf{v}_j)$. In other words, we need to verify the following.

$$\log_{P_2} C_1 = \log_{W_\upsilon} C_2 \text{ and } \log_{P_2} C_1 = \log_{V_1(\mathsf{v}_1)} B_1 = \cdots = \log_{V_j(\mathsf{v}_j)} B_j.$$

In Figure 1, the first equality is explicitly verified, whereas the second equality is not. The idea is that if the second equality does not hold, then the key K that will be reconstructed will be improper and indistinguishable from random (to the adversary). Correspondingly, the quantities (IV, dk) will also be indistinguishable from random and symmetric authentication with this pair will fail (otherwise the adversary has broken the authentication of the AE scheme). Thus, instead of using j pairings for verifying the second equality, we use symmetric authentication to reject invalid ciphertext. This leads to a more efficient decryption algorithm. Note that the use of hybrid encryption is very crucial in the current context. This is similar to the Kurosawa-Desmedt PKE (Kurosawa and Desmedt, 2004), which provides improved efficiency over the Cramer-Shoup scheme for hybrid encryption.

The additional requirements of group elements and operations for attaining CCA-security compared to the scheme in Section 3.1 consists of the following.

1 one extra element $W \in \mathbb{G}_1$ in the public parameters

2 two additional scalar multiplications in $\mathbb{G}_1$ during encryption

3 one additional scalar multiplication in $\mathbb{G}_1$ and one pairing-based verification during decryption.

6.1 Security statement

The security statement for the new scheme is given below.

Theorem 4. The HIBE scheme described in Figure 1 is $(\epsilon_{hibe}, t, q_{\mathsf{ID}}, q_C)$-CCA secure assuming that the (ϵ_{dbdh}, t')-DBDH assumption holds in $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$; H_s is an (ϵ_{uowhf}, t')-UOWHF; KDF is (ϵ_{kdf}, t')-secure; and the AE scheme possesses (ϵ_{auth}, t')-authentication security and (ϵ_{enc}, t') one-time encryption security; where

$$\begin{aligned}\epsilon_{hibe} \leq \tfrac{q}{p} + \epsilon_{uowhf} + 2h(m(\mu_l + 1))^h \epsilon_{dbdh} \\ +hq_C\epsilon_{auth} + 2\epsilon_{kdf} + \epsilon_{enc};\end{aligned} \tag{36}$$

$t' = t + t_{sim}$ and t_{sim} is the simulation time, i.e., the time to generate q_{ID} private keys, decrypt q_C ciphertexts and generate one ciphertext plus a time of $O(\epsilon_{hibe}^{-2} \ln(\epsilon_{hibe}^{-1})\lambda^{-1} \ln(\lambda^{-1}))$; $\mu_l = l(2^{n/l} - 1)$; $m = \max(2q, 2^{n/l})$ and $q = q_{\mathsf{ID}} + q_C$. We further assume $m(1 + \mu_l) < p$.

The proof is given in Section 6.2. The statement of Theorem 4 is almost the same as that of Theorem 1 with the following differences.

1 The above theorem states CCA-security.

2 The security degradation of ϵ_{dbdh} is by a factor of $2h(m(\mu_l + 1))^h$ in the above statement where as it is equal to $2(m(\mu_l + 1))^h$ in Theorem 1, i.e., there is an additional degradation by a factor of h.

3 The value of q in the expression for m is the sum of q_{ID} and q_C whereas in Theorem 1 it is only q_{ID}. The reason for having q_C as part of q is that it may be required to simulate decryption queries using key extraction queries.

For $2q \geq 2^{n/l}$ (typically l would be chosen to ensure this), we have

$$\begin{aligned}\epsilon_{hibe} \leq \epsilon_{uowhf} + \ 2h(2lq2^{n/l})^h \epsilon_{dbdh} \\ +2\epsilon_{kdf} + \epsilon_{enc} + hq_C\epsilon_{auth}.\end{aligned}$$

The corresponding upper bound on ϵ_{hibe} from Theorem 1 is $(2lq_{\mathsf{ID}}2^{n/l})^h\epsilon_{dbdh}$. Thus, we get an additional security degradation by a factor of h while attaining CCA-security. Since h is the maximum number of levels in the HIBE, its value is small and the degradation is not significant. Also, q in the present case includes both key extraction and decryption queries.

6.2 Proof of Theorem 4

The construction of CCA-secure HIBE in Figure 1 is built on the construction of CPA-secure HIBE given in Section 3.1. The proof of Theorem 1 shows how to set-up the scheme, answer key-extraction queries and generate the challenge ciphertext. The proof of Theorem 4 incorporates these aspects of the proof of Theorem 1. Additionally, we have the following considerations.

1 definition of W during set-up.

2 generation of C_2 during challenge generation as well as generation of a proper ciphertext using the AE scheme.

3 properly answering decryption queries.

The proof of Theorem 4 is given as a sequence of games. In each game, a bit γ is chosen randomly and the adversary makes a guess γ'. By X_i we denote the event that $\gamma = \gamma'$ in the i^{th} game.

Game 0. This is the usual adversarial game for defining CCA-security of HIBE schemes. We assume that the adversary's runtime is t, it makes q_{ID} key-extraction queries and q_{C} decryption queries. Also, we assume that the adversary maximises the advantage among all adversaries with similar resources. Thus, we have

$$\epsilon_{hibe} = \left|\Pr[X_0] - \frac{1}{2}\right|.$$

The group element C_1^* which is provided to the adversary during the challenge generation does not depend on the adversary's input. We will assume that this is randomly chosen during setup. Also, we will assume that during set-up an integer h_θ is chosen uniformly at random from $\{1, \ldots, h\}$. The significance of h_θ will become clear later. We will denote the quantities corresponding to the challenge by a superscript $*$.

Game 1. This is the same as Game 0, with the following change. If the adversary ever submits a decryption query of the form $(C_1, C_2, B_1, \ldots, B_j)$ with $(j, C_1) \neq (h_\theta, C_1^*)$ and $H_s(j, C_1) = H_s(h_\theta, C_1^*)$, then the simulator rejects the query. Let F_1 be the event that a decryption query is rejected only by this check. It is easy to see that $\Pr[F_1] \leq \epsilon_{uowhf}$. If F_1 does not occur, then Game 0 and Game 1 are identical. Using the difference lemma (as named in Shoup, 2004), we obtain

$$|\Pr[X_0] - \Pr[X_1]| \leq \Pr[F_1] \leq \epsilon_{uowhf}.$$

Game 2. This game is the main non-trivial game of the proof and is based on Game 1 in the proof of Theorem 1. The scheme is setup from a tuple $(aP_1, aP_2, bP_1, cP_1, cP_2, Z = e(P_1, P_2)^{abc})$, where we assume that a, b and c are known to the simulator. There are four parts to this game – setup; simulation of key-extraction queries; simulation of decryption queries; and challenge generation.

As in Game 1 in the proof of Theorem 1, for certain queries as well as for certain challenge identities, the simulator is unable to answer without using the values of a, b or c. In such cases, it sets a flag flg to 1 (which is initially set to 0). However, it always answers the adversary's queries properly and hence the adversary's view remains unchanged from the previous game. Thus, we have $\Pr[X_1] = \Pr[X_2]$.

Set-up. Set $Q_2 = aP_2$ and $R_1 = bP_1$. The secret key is $aR_1 = abP_1$. Also, set $C_1^* = cP_2$ and h_θ is chosen during set-up as mentioned in Game 0.

The public parameters $(U'_1, \ldots, U'_h, U_1, \ldots, U_l)$ are required to handle key extraction queries. The construction of these parameters are as in the proof of Theorem 1.

The parameter W is required for answering decryption queries (and is not present in the proof of Theorem 1). We show how to define W. Compute $\upsilon = H_s(h_\theta, cP_2)$; choose β randomly from $\mathbb{Z}_p$ and define $W = -\upsilon bP_1 + \beta P_1$. The choice of h_θ corresponds to the fact that at this point we are guessing the length of the challenge identity.

Key extraction query. The technique for answering such queries is as described in the proof of Theorem 1.

Decryption query. Suppose $C = (C_1, C_2, B_1, \ldots, B_j)$ is a decryption query for the identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$. There are several cases to consider.

Case $(\mathsf{v}_1, \ldots, \mathsf{v}_j)$ *is not a prefix of* $(\mathsf{v}_1^*, \ldots, \mathsf{v}_{h_\theta}^*)$*:.* In this case, a private key d_v for v is obtained using the technique for simulating key extraction query. This d_v is used to decrypt the ciphertext. In the process of key extraction, the variable flg might have to be set to one.

Case $(\mathsf{v}_1, \ldots, \mathsf{v}_j)$ *is a prefix of* $(\mathsf{v}_1^*, \ldots, \mathsf{v}_{h_\theta}^*)$*:.* If either $j < h_\theta$ or $C_1 \neq C_1^*$, then by Game 1, we can assume that $H_s(j, C_1) \neq H_s(h_\theta, C_1^*)$. So suppose that $(j, C_1) = (h_\theta, C_1^*)$. We assume that this happens only in Phase 2. In Phase 1, the randomly chosen C_1^* is not available to the adversary and hence the event $C_1 = C_1^*$ can occur only with negligible probability of q/p, which accounts for the term q/p in the security bound.

Using $j = h_\theta$, we have $(\mathsf{v}_1, \ldots, \mathsf{v}_j) = (\mathsf{v}_1^*, \ldots, \mathsf{v}_{h_\theta}^*)$. Recall that $C_1^* = cP_2$ and so $C_1 = C_1^*$ implies that $C_1 = cP_2$, i.e., $\log_{P_2}(C_1) = c$. Now $C_1^* = cP_2$ implies that $B_i^* = cV_i^{(1)}(\mathsf{v}_i^*)$. Also, $\mathsf{v}_i = \mathsf{v}_i^*$ implies $V_i^{(1)}(\mathsf{v}_i) = V_i^{(1)}(\mathsf{v}_i^*)$. This and $\log_{P_2}(C_1) = c$ implies $B_i = cV_i^{(1)}(\mathsf{v}_i) = cV_i^{(1)}\ (\mathsf{v}_i^*) = B_i^*$, as otherwise the query is necessarily mal-formed and is rejected. As a result, $(B_1, \ldots, B_j) = (B_1^*, \ldots, B_j^*)$. Also, using $(j, C_1) = (h_\theta, C_1^*)$ it is easy to verify that $C_2 = C_2^*$. Thus, we have $(C_1, C_2, B_1, \ldots, B_j) = (C_1^*, C_2^*, B_1^*, \ldots, B_j^*)$. In other words, the decryption query is on the challenge ciphertext, which is not allowed in the game. Hence, we cannot have $(j, C_1) = (h_\theta, C_1^*)$ and so $H_s(j, C_1) \neq H_s(h_\theta, C_1^*)$.

Let $\upsilon' = H_s(j, C_1)$ and $W_{\upsilon'} = W + \upsilon' bP_1$. The simulator verifies whether $e(W_{\upsilon'}, C_1) = e(C_2, P_2)$ and proceeds if the test succeeds. If the test fails, it returns $\perp$ to $\mathcal{A}$. Note that, at this point, since we have verified that $e(W_{\upsilon'}, C_1) = e(C_2, P_2)$, we can write $C_1 = tP_2$ and $C_2 = tW_{\upsilon'}$ for some t in $\mathbb{Z}_p$.

Choose r randomly from $\mathbb{Z}_p$ and compute $E_{\upsilon'}$ and $d_{\upsilon'}$ in the following manner. Recall that $\upsilon = H_s(h_\theta, cP_2)$ and $W = -\upsilon bP_1 + \beta P_1$. Since, $\upsilon' = H_s(j, C_1) \neq H_s(h_\theta, C_1^*) = \upsilon$, the inverse of $(\upsilon' - \upsilon)$ (modulo p) exists.

$$\left.\begin{aligned} E_{\upsilon'} &= \tfrac{-\beta}{\upsilon'-\upsilon} aP_1 + r((\upsilon' - \upsilon)bP_1 + \beta P_1) \\ &= abP_1 + \left(r - \tfrac{a}{\upsilon'-\upsilon}\right)(\upsilon' bP_1 + W) \\ &= abP_1 + \widetilde{r} W_{\upsilon'} \\ d_{\upsilon'} &= rP_2 - \tfrac{1}{\upsilon'-\upsilon} aP_2 \\ &= \widetilde{r} P_2. \end{aligned}\right\} \tag{37}$$

This technique is based on Boneh et al. (2005b) which is in turn based on the technique of Boneh and Boyen (2004a). The verification of the above computation is quite routine – in particular the second equality can be easily seen by substituting $W = -\upsilon bP_1 + \beta P_1$.

The decryption can now be performed as follows.

$$\begin{aligned} \frac{e(E_{\upsilon'}, C_1)}{e(C_2, d_{\upsilon'})} &= \frac{e(abP_1 + \widetilde{r}W_{\upsilon'}, tP_2)}{e(tW_{\upsilon'}, \widetilde{r}P_2)} \\ &= e(R_1, Q_2)^t. \end{aligned}$$

Note that any such decryption query can be answered without using the values of a, b or c.

The simulation of the decryption query makes the role of W clear. The scheme uses $K = e(Q_1, R_2)^t$ to be the encapsulated secret key and creates two encapsulations of it. The first encapsulation is using the HIBE scheme of Section 3.1, where as the second encapsulation is using the BB-IBE scheme from Boneh and Boyen (2004a). In the actual scheme, the second encapsulation is never used (apart from verifying its correctness). It is used in the simulation to obtain K and answer a decryption query if the identity of the decryption query is a prefix of the challenge identity. The advantage is that the BB-IBE scheme is only required to be selective-ID secure and hence the ‘challenge identity’ υ for the BB-IBE scheme can be generated during set-up.

Challenge. The adversary submits a challenge identity $(\mathsf{v}_1^*, \ldots, \mathsf{v}_{h^*}^*)$ and two M_0 and M_1 of equal lengths. The challenge ciphertext is of the form $(C_1^*, C_2^*, B_1^*, \ldots, B_{h^*}^*)$, where we have already chosen $C_1^* = cP_2$ during set-up. The components B_1^* to $B_{h^*}^*$ are generated as in the proof of Theorem 1.

The component C_2^* is new to this scheme and we show how to generate it. If $h^* \neq h_\theta$, then set flg to 1, i.e., the random guess of the length of the challenge identity during the set-up turns out to be incorrect. In this case, the simulator uses a, b and c to generate the challenge and answer the adversary. Otherwise, set $C_2^* = \beta cP_1$. This C_2^* is properly formed. To see this first note that $\upsilon = H_s(h^*, cP_2)$ and so we require $C_2^* = cW_\upsilon$. This follows from the following calculation.

$$\begin{aligned} C_2^* &= cW_\upsilon = c(\upsilon bP_1 + W) \\ &= c(\upsilon bP_1 - \upsilon bP_1 + \beta P_1) = c\beta P_1 = \beta cP_1. \end{aligned}$$

Choose a random bit γ. Set $K^* = Z$ and then apply the rest of the encryption procedure to complete the encryption for the message M_γ.

Game 3. This game is the same as Game 2, with the only difference that the Z in Game 2 is now replaced by a random element of $\mathbb{G}_T$.

Details of how to obtain a DBDH solver from the two games are given as part of the proof of Theorem 1. This analysis also holds for the current proof. The only new abort condition is during challenge generation, when $h^* \neq h_\theta$. Since $1 \leq h^*, h_\theta \leq h$ and h_θ is chosen randomly from $\{1, \ldots, h\}$, the probability of this new abort is $1/h$. With this small change, an analysis similar to the one done in the proof of Theorem 1 shows the following result.

Proposition 6. $|\mathsf{Pr}[X_2] - \mathsf{Pr}[X_3]| \leq 2h(m(\mu_l + 1))^h \epsilon_{dbdh}$.

Game 4. At this point, we have K^* to be random. Since we are assuming that a, b and c are known to the simulator, we can also assume that u'_j and u_i are known to the simulator such that $U'_j = u'_j P_1$ and $U_i = u_i P_1$. This follows easily from the definition of U_i and U'_j given in (9) in the proof of Theorem 1. More explicitly, $u'_j = (p - mk_j + x'_j) b + y'_j$ and $u_i = x_i b + y_i$.

Knowing the u'_js and the u_is and using the definition of V_i in (2) we can assume that for any V_i, the simulator is able to compute w_i such that $V_i = w_i P_1$. The adversary may submit a decryption query with $C_1 = tP_2$ and for some i, $B_i = t_1 V_i$ with $t \neq t_1$. The knowledge of w_i allows the simulator to test for this in the following manner: If $e(V_i, C_1) \neq e(w_i C_1, P_2)$, then $t_1 \neq t$ and the query is malformed. The simulator can now detect and reject such a query. Note that this checking is not done in the actual scheme. So, we would like to be assured that the probability of getting to this checking stage is small. In other words, we would like to be assured that if the query is malformed as above and the scheme does not reject it, then the adversary has broken the authentication property of the AE scheme.

Let *Rejection Rule 0* be the rule whereby a ciphertext is rejected based on the failure of the authentication property of the AE scheme. Let *Rejection Rule 1* be the rejection rule mentioned above. Let F_4 be the event that a malformed query is rejected by *Rule 1* but not by *Rule 0*. Our aim is to show that the probability of this happening is low. Note that if no query is rejected by *Rule 1*, then Games 3 and 4 are identical.

From this point onwards, we will only be considering decryption queries. The adversary makes a total of q_C decryption queries. We will use the superscript $(\jmath)$ to denote the quantities related to the $\jmath^{\text{th}}$ decryption query. For example, $K^{(\jmath)}$ denotes the input to $\mathsf{KDF}()$ in the $\jmath^{\text{th}}$ decryption query.

We now employ a 'plug and pray' technique used in Kurosawa and Shoup (2005) and assume that the $\imath^{\text{th}}$ component of the $\jmath^{\text{th}}$ query is malformed, i.e., $C_1^{(\jmath)} = tP_2$ and $B_\imath^{(\jmath)} = t_1 V_\imath(\mathsf{v}_\imath^{(\jmath)})$ with $t \neq t_1$. Note that the 'plug and pray' here also extends over the levels of the HIBE, a feature which is not required in Kurosawa and Shoup (2005). Let F'_4 be the event that the query is not rejected by *Rule 0* but the $\imath^{\text{th}}$ component of the $\jmath^{\text{th}}$ query fails *Rule 1*. Then $\mathsf{Pr}[F_4] \leq h \times q_C \times \mathsf{Pr}[F'_4]$ and we have

$$|\mathsf{Pr}[X_3] - \mathsf{Pr}[X_4]| \leq \mathsf{Pr}[F_4] \leq h \times q_{\mathsf{C}} \times \mathsf{Pr}[F'_4]. \tag{38}$$

We would like to upper bound $\mathsf{Pr}[F'_4]$. For this we use the deferred analysis technique of Kurosawa and Shoup (2005). Also, since we have done a 'plug and pray' over the levels of the HIBE, henceforth we will assume that there is only one level in the HIBE, i.e., we are considering an IBE scheme. This will simplify the notation as this will result in only one B which is of the form tV with $V = wP_1$.

Game 5. We modify Game 4 in the following manner. If the $\jmath^{\text{th}}$ decryption query is detected to be malformed using *Rule 1*, then we set $K^{(\jmath)}$ to be a random element of $\mathbb{G}_2$. We now have to argue that this does not change the adversary's point of view. In effect, we are setting both K^* and $K^{(\jmath)}$ to be independent random elements and have to argue that this is what the adversary can expect to see.

Let us now analyse the relationship between the identity v^* for the challenge ciphertext and the identity $\mathsf{v}^{(\jmath)}$ for the malformed query. There are two cases to consider.

Case $\mathsf{v}^* = \mathsf{v}^{(\jmath)}$. In this case, the adversary cannot ask for the private key of $\mathsf{v}^{(\jmath)}$. Let the secret key corresponding to $\mathsf{v}^{(\jmath)}$ be $(abP_1 + rV^{(\jmath)}, rP_2)$, where r is a random element of $\mathbb{Z}_p$. Then the adversary expects $K^{(\jmath)}$ of the malformed query to be

$$\begin{aligned} K^{(\jmath)} &= \frac{e(abP_1 + rV^{(\jmath)}, tP_2)}{e(t_1 V^{(\jmath)}, rP_2)} \\ &= e(bP_1, aP_2)^t \times e(P_1, P_2)^{wr(t-t_1)}. \end{aligned}$$

Since $t \neq t_1$ (as the query is malformed) and r is uniform random, $K^{(\jmath)}$ is also uniform random. On the other hand, the adversary expects K^* to be $e(bP_1, aP_2)^{t^*}$ where t^* is uniform random. Hence, the adversary expects K^* to be random. Further, the randomness of $K^{(\jmath)}$ and K^* depend on the randomness of r and t^* which are independent. Hence, the adversary also expects $K^{(\jmath)}$ and K^* to be independent and uniform random quantities as provided to the adversary.

Case $\mathsf{v}^* \neq \mathsf{v}^{(\jmath)}$. In this case, the adversary can ask for the secret key for $\mathsf{v}^{(\jmath)}$ but not before making the malformed decryption query. If the adversary knows the secret key for $\mathsf{v}^{(\jmath)}$, then he can decrypt any ciphertext encrypted using $\mathsf{v}^{(\jmath)}$. Thus, it is useless for him to query the decryption oracle using $\mathsf{v}^{(\jmath)}$ *after* obtaining the secret key for $\mathsf{v}^{(\jmath)}$. Recall that we had disallowed such useless queries.

The adversary can first ask for the decryption of a malformed query and then ask for the private key for the same identity. We have to ensure that the answers to the decryption and private key queries are consistent. By consistency we mean the following. Suppose the adversary makes a decryption query with $\mathsf{v}^{(\jmath)}$ and later a private key extraction query on $\mathsf{v}^{(\jmath)}$. With the private key $d_{\mathsf{v}^{(\jmath)}}$ returned to him, the adversary can decrypt its own earlier decryption query. Consistency requires that the output given to him on his decryption query should be equal to what he computes for himself. The next modification ensures this consistency. In this case, the independence of K^* and $K^{(\jmath)}$ will be easily ensured.

Let the $\jmath^{\text{th}}$ query be of the form $(t^{(\jmath)}P_2, t_1^{(\jmath)}V)$. Suppose the simulator returns $K^{(\jmath)} = e(bP_1, aP_2)^{t_2^{(\jmath)}}$. On a later private key query on $\mathsf{v}^{(\jmath)}$, the simulator has to return $(abP_1 + r^{(\jmath)}V, r^{(\jmath)}P_2)$ for some uniform random $r^{(\jmath)} \in \mathbb{Z}_p$. The consistency requirement is satisfied if

$$\begin{aligned} e(P_1, P_2)^{abt_2^{(\jmath)}} &= e(bP_1, aP_2)^{t_2^{(\jmath)}} = K^{(\jmath)} \\ &= \frac{e(abP_1 + r^{(\jmath)}V, t^{(\jmath)}P_2)}{e(t_1^{(\jmath)}V, r^{(\jmath)}P_2)} \\ &= e(bP_1, aP_2)^{t^{(\jmath)}} \times e(V, P_2)^{r^{(\jmath)}(t^{(\jmath)} - t_1^{(\jmath)})} \\ &= e(P_1, P_2)^{abt^{(\jmath)} + wr(t^{(\jmath)} - t_1^{(\jmath)})}. \end{aligned}$$

Recall that $V = wP_1$. The above consistency condition can be written as

$$t_2^{(\jmath)} = t^{(\jmath)} + \frac{wr^{(\jmath)}(t^{(\jmath)} - t_1^{(\jmath)})}{ab}. \tag{39}$$

Note that the simulator does not know $t^{(\jmath)}$ and $t_1^{(\jmath)}$. The $\jmath^{\text{th}}$ malformed query is answered in the following manner. The simulator chooses an $r^{(\jmath)}$ (required for answering a possible future key extraction query on $\mathsf{v}^{(\jmath)}$) uniformly at random; computes $A = e(P_1, P_2)^{abt^{(\jmath)}}$ (this can be done since the simulator knows a, b and $t^{(\jmath)}P_2$) and then computes

$$\begin{aligned} B &= \frac{e(r^{(\jmath)}V^{(\jmath)}, t^{(\jmath)}P_2)}{e(t_1^{(\jmath)}V^{(\jmath)}, r^{(\jmath)}P_2)} \\ &= e(P_1, P_2)^{r^{(\jmath)}w(t^{(\jmath)} - t_1^{(\jmath)})}. \end{aligned}$$

Note that both numerator and denominator is computable from what is known to the simulator. Then the simulator computes $K^{(\jmath)} = (A \times B)^{1/(ab)}$ which is equal to $e(bP_1, aP_2)^{t_2^{(\jmath)}}$ where $t_2^{(\jmath)}$ is as given in (39). This value $K^{(\jmath)}$ is returned to the adversary. Since $r^{(\jmath)}$ is random, so is $t_2^{(\jmath)}$ and hence $K^{(\jmath)}$ is random. Later if the adversary asks for the private key for $\mathsf{v}^{(\jmath)}$, then the simulator uses $r^{(\jmath)}$ to construct the private key and answer the adversary.

Define F_5' in a manner similar to F_4'. Then we have

$$\mathsf{Pr}[X_4] = \mathsf{Pr}[X_5] \text{ and } \mathsf{Pr}[F_4'] = \mathsf{Pr}[F_5']. \tag{40}$$

Game 6. This is obtained from Game 5 by the following modification. In Game 5, the keys $(\mathsf{IV},^* dk^*)$ and $(\mathsf{IV}^{(\jmath)}, dk^{(\jmath)})$ are obtained by applying KDF to K^* and $K^{(\jmath)}$ respectively. In Game 6, these are generated randomly. Define F_6' in a manner similar to that of F_4'. Then we have

$$|\mathsf{Pr}[X_5] - \mathsf{Pr}[X_6]| \le 2\epsilon_{kdf} \text{ and } |\mathsf{Pr}[F_5'] = \mathsf{Pr}[F_6']|. \tag{41}$$

The factor of two comes due to the fact that the adversary can break one out of these two invocations of KDF.

In Game 6, K^* and (IV, dk) are random and independent of the elements $C_1^*, C_2^*, B_1^*, \ldots, B_{h^*}^*$. The message M_γ is encrypted using the random (IV^*, dk^*). If the adversary is able to correctly guess γ, then the one-time security of the AE scheme is broken. Hence, $|\mathsf{Pr}[X_6] - 1/2| \le \epsilon_{enc}$.

Now, we turn to bounding the probability of the event F_6'. Recall that the occurrence of the event F_6' implies that the query has passed the authentication of the underlying AE scheme. At this point, we have $K^\jmath$ to be uniform random and hence, using the security of KDF, the pair $(\mathsf{IV}^\jmath, dk^\jmath)$ is also uniform random (and unknown to the adversary). Thus, the adversary has been able to obtain a forgery for the AE scheme under a uniform random key (without even making any previous queries for this key). This violates the authentication property of the AE scheme and hence $\mathsf{Pr}[F_6'] \le \epsilon_{auth}$. Finally, combining all the inequalities, we obtain

$$\begin{aligned} \epsilon_{hibe} &= \left| \mathsf{Pr}[X_0] - \frac{1}{2} \right| \\ &\le |\mathsf{Pr}[X_0] - \mathsf{Pr}[X_1]| + |\mathsf{Pr}[X_2] \\ &\quad - \mathsf{Pr}[X_3]| + |\mathsf{Pr}[X_3] - \mathsf{Pr}[X_4]| \\ &\quad + |\mathsf{Pr}[X_5] - \mathsf{Pr}[X_6]| + |\mathsf{Pr}[X_6] - 1/2| \\ &\le q/p + \epsilon_{uowhf} + 2h(m(\mu_l + 1))^h \epsilon_{dbdh} \\ &\quad + hq_C\epsilon_{auth} + 2\epsilon_{kdf} + \epsilon_{enc}. \end{aligned}$$

This completes the proof. □

6.3 *Comparison to previous work*

The CCA-secure HIBE scheme constructions described in the previous section can be specialised to obtain CCA-secure PKE and IBE as special cases. We show that when specialised to a PKE scheme, one gets the BMW construction in the Type 3 setting. When specialised to IBE schemes, we obtain more efficient IBE schemes compared to the previously best known constructions.

Public key encryption. In this case there are no identities and no PKG. It is possible to make the following simplifications.

SetUp:

1. the elements $U_1', \ldots, U_h', U_1, \ldots, U_l$ are no longer required
2. the UOWHF H_s can be replaced by an injective embedding from $\mathbb{G}_2$ to $\mathbb{Z}_p$
3. a random w in $\mathbb{Z}_p$ is chosen and W is set to be equal to wP_1
4. a random $\alpha \in \mathbb{Z}_p$ is chosen and R_1 is set to be equal to αP_1
5. the secret key is now $(\alpha Q_2, \alpha, w) \in \mathbb{G}_2 \times \mathbb{Z}_p^2$ while the public key is $(P_1, W, e(R_1, Q_2)) \in \mathbb{G}_1^2 \times \mathbb{G}_T$

6 the AE scheme can be replaced with a one-time secure data encapsulation mechanism (DEM)

7 the public key consists of two elements of $\mathbb{G}_1$ and a single element of $\mathbb{G}_T$. The secret key, on the other hand, consists of a single element of $\mathbb{G}_2$ and two elements of $\mathbb{Z}_p$.

KeyGen: This is not required at all.

Encrypt:

1 set $C_1 = tP_1$. Note that C_1 is now an element of $\mathbb{G}_1$ which reduces the ciphertext overhead

2 the elements $B_1, \ldots, B_j$ are not required

3 encryption with a DEM will not produce a tag.

4 the ciphertext expansion consists of two elements of $\mathbb{G}_1$

5 one encryption takes time $3[\mathrm{SM}_1] + 1[\mathrm{E}]$.

Decrypt:

1 The purpose of the pairing verification in (H)IBE was to ensure that the same randomiser t is used in the computation of C_1 and C_2.from 1 to h. For an IBE, the length is always one. Hence, in this Recall that $C_1 = tP_1$ and $C_2 = tW_v$, where $W_v = W + vR_1$. With the knowledge of w and α, this can be done as follows. Compute $w' = w + v\alpha$ and verify whether $w'C_1 = C_2$. This requires only one scalar multiplication in $\mathbb{G}_1$ as opposed to one pairing verification. It is also due to such verification that we are able to work with C_2 in $\mathbb{G}_1$ which follows from W being in $\mathbb{G}_1$. The HIBE in Figure 1 has $W \in \mathbb{G}_2$ and hence C_2 also in $\mathbb{G}_2$.

2 The value of K is reconstructed as $K = e(C_1, \alpha Q_2)$.

3 Since the AE scheme is replaced with a DEM, symmetric authentication will not be done.

4 The time for decryption is $1[\mathrm{SM}_1] + 1[\mathrm{P}_1]$.

With these simplifications, we obtain the BMW scheme in the Type 3 pairing setting. The details are shown in Figure 2. A security reduction for this scheme can be extracted from the proof of Theorem 4 and will be the Type 3 counter-part of the proof in Boneh et al. (2005b). The efficiency of the Type 3 variant of BMW is comparative with the well-known (Kurosawa and Desmedt, 2004) PKE scheme.

Identity-based encryption. In this case $h = 1$. The scheme in Figure 1 remains unchanged except for one simplification. In a HIBE, the length of the identity tuple can vary from 1 to h. For an IBE, the length is always one. Hence, in this case, we can restrict the domain of H_s to be $\mathbb{G}_2$. Since, $\mathbb{G}_2$ has cardinality p, the domain and range of H_s are the same and we can also take H_s to be an injective embedding from $\mathbb{G}_2$ to $\mathbb{Z}_p$ as has been done in the BMW construction.

Let us denote the IBE scheme arising from the HIBE scheme in Figure 1 to be IBE-1. This corresponds to converting the (CPA-secure) HIBE-1 scheme in Section 3.1 to a CCA-secure HIBE scheme and then instantiating that to an IBE by putting $h = 1$ (except for doing away with the length as an input to H_s). As mentioned earlier, one can similarly convert HIBE-2 of Section 3.2 to obtain a CCA-secure HIBE scheme. For this scheme, when $h = 1$ there is an additional optimisation that is possible where the public key size can be made equal to that of IBE-1 at the cost of increasing the size of the master secret key. In Figure 3 we provide the details of this IBE scheme which we call IBE-2.

Figure 2 CCA-secure PKE scheme

PKE.SetUp 1. Choose α randomly from $\mathbb{Z}_p$; 2. Choose w randomly from $\mathbb{Z}_p$; 3. Set $R_1 = \alpha P_1$; $W = \alpha P_1$. 4. Choose Q_2 randomly from $\mathbb{G}_2$. 5. Let $H_s : \mathbb{G}_2 \to \mathbb{Z}_p$ be chosen from a UOWHF and made public. 6. Public key: $e(R_1, Q_2), W, P_1$. 7. Secret key: $\alpha Q_2, \alpha, w$.	PKE.Encrypt message M. 1. Choose t randomly from $\mathbb{Z}_p$. 2. $C_1 = tP_1$, 3. $K = e(R_1, Q_2)^t$. 4. $dk = \mathsf{KDF}(K)$. 5. $(\mathsf{cpr}, \mathsf{tag}) = \mathsf{DEM.Encrypt}_{dk}(M)$. 6. $v = H_s(C_1)$; $W_v = W + vR_1$; $C_2 = tW_v$. 7. Output (C_1, C_2, cpr). PKE.Decrypt: ciphertext (C_1, C_2, cpr); secret key $\alpha Q_2, \alpha, w$. 1. $v = H_s(C_1)$; $w' = w + v\alpha$; 2. If $w'C_1 \neq C_2$ return $\perp$. 3. $K = e(C_1, \alpha Q_2)$. 4. $dk = \mathsf{KDF}(K)$. 5. $M = \mathsf{DEM.Decrypt}_{dk}(C)$. 6. Output M.

Note: The pairing setting $(p, \mathbb{G}_1 = \langle P_1 \rangle, \mathbb{G}_2 = \langle P_2 \rangle, \mathbb{G}_T, e)$ is as defined in Section 2.3.

Figure 3 IBE-2: a CCA-secure IBE scheme

<u>IBE.SetUp</u> 1. Choose α randomly from $\mathbb{Z}_p$; 2. Set $R_1 = \alpha P_1$; $R_2 = \alpha P_2$. 3. Choose Q_2 randomly from $\mathbb{G}_2$. 4. Choose $x, y_1, \ldots, y_l$ randomly from $\mathbb{Z}_p$. 5. Set $U' = xP_1, U_1 = y_1 P_1, \ldots, U_l = y_l P_1$. 6. Choose w randomly from $\mathbb{Z}_p$; 7. Set $W_1 = wP_1$; $W_2 = wP_2$; 8. Let $H_s : \mathbb{G}_2 \to \mathbb{Z}_p$ be chosen from a UOWHF and made public. 9. Public parameters: R_2, W_2 (required only for decryption) $R_1, P_1, W_1, e(R_1, Q_2), U', U_1, \ldots, U_l$. 10. Master secret key: $\alpha Q_2, x, y_1, \ldots, y_l$.	<u>IBE.KeyGen</u>: Identity v. 1. Choose r randomly from $\mathbb{Z}_p$. 2. $d_0 = \alpha Q_2 + r\left(x + \sum_{i=1}^{l} \mathsf{v}_i y_i\right) P_2$. 3. $d_1 = rP_2$. 4. Output $d_\mathsf{v} = (d_0, d_1)$.
<u>IBE.Encrypt</u>: Identity v; message M. 1. Choose t randomly from $\mathbb{Z}_p$. 2. $C_1 = tP_1$, $B_1 = tV^{(1)}(\mathsf{v})$. 3. $K = e(R_1, Q_2)^t$. 4. $(\mathsf{IV}, dk) = \mathsf{KDF}(K)$. 5. $(\mathsf{cpr}, \mathsf{tag}) = \mathsf{AE.Encrypt}_{dk}(\mathsf{IV}, M)$. 6. $\upsilon = H_s(C_1)$; $W_\upsilon = W_1 + \upsilon R_1$; $C_2 = tW_\upsilon$. 7. Output $(C_1, C_2, B_1, \mathsf{cpr}, \mathsf{tag})$.	<u>IBE.Decrypt</u>: Identity v; ciphertext $(C_1, C_2, B_1, \mathsf{cpr}, \mathsf{tag})$; decryption key $d_\mathsf{v} = (d_0, d_1)$. 1. $\upsilon = H_s(C_1)$; $W'_\upsilon = W_2 + \upsilon R_2$. 2. If $e(C_1, W'_\upsilon) \neq e(C_2, P_2)$ return $\perp$. 3. $K = e(d_0, C_1) \times e(B_1, -d_1)$. 4. $(\mathsf{IV}, dk) = \mathsf{KDF}(K)$. 5. $M = \mathsf{AE.Decrypt}_{dk}(\mathsf{IV}, C, \mathsf{tag})$. (This may abort and return $\perp$). 6. Output M.

Notes: 1 Identities are n-bit strings and an identity v is written as $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_l)$ where v_i is an (n/l)-bit string.
2 The pairing setting $(p, \mathbb{G}_1 = \langle P_1 \rangle, \mathbb{G}_2 = \langle P_2 \rangle, \mathbb{G}_T, e)$ is as defined in Section 2.3.
3 The notation $V_k^{(1)}$ and $V_k^{(2)}$ are as given in (2) and (3) respectively.

A proof of security for IBE-2 can be described quite easily along the lines of Theorem 4. The hard problem would be the DBDH-3b problem described in Section 2.4 (in fact, the security of HIBE-2 can also be based on the hardness of the DBDH-3b problem). In the simulation, we set $Q_2 = aP_2$, $R_1 = bP_1$, $R_2 = bP_2$, $C_1^* = cP_1$, $W_1 = -\upsilon(bP_1) + \beta P_1$ and $W_2 = -\upsilon(bP_2) + \beta P_2$.

Let us now compare IBE-1 and IBE-2 with the previous construction by Kiltz and Galindo (KG) (2009). This comparison is given in Table 6. For IBE-2 (as shown in Figure 3), in Table 6, we have included R_2 and W_2 as part of the decryption key and not as part of the public parameters. These two quantities are not required for encryption and in real implementations would be passed to a user along with the decryption key.

Kiltz and Galindo (2009) suggests a method of implicit rejection. They construct an identity-based key encapsulation mechanism such that the following holds. If the ciphertext is valid (i.e., the ciphertext has been produced by invoking the encapsulation algorithm), then the proper key is generated, while if the ciphertext is invalid, then a random key is generated. When combined with a one-time secure DEM, the decryption algorithm of the corresponding IBE scheme never rejects any ciphertext. This results in replacing some of the pairing computations by scalar multiplications.

In a work subsequent to the publication of our conference paper (Sarkar and Chatterjee, 2007), Kiltz and Vahlis (KV) (2008) use symmetric authentication techniques (akin to the techniques used in Sarkar and Chatterjee, 2007) and a different hardness assumption to obtain an IBE having improved efficiency of encryption and decryption. The assumption they use is the one used in Boneh et al. (2005a) and Chatterjee and Sarkar (2006b) tailored to work for IBE: given P, aP, bP, b^2P, cP and Z, determine whether Z is equal to $e(P,P)^{abc}$ or whether Z is random. This is called the mBDDH assumption in Kiltz and Vahlis (2008). In comparison to the more usual DBDH assumption, in this case, the extra element b^2P is provided. The more general version of this assumption was introduced in Boneh et al. (2005a) where b^iP for several more values of i are provided as part of the problem instance. Coming back to the mBDDH assumption, the extra element b^2P allows the proof of the scheme in Kiltz and Vahlis (2008) to simulate the generation of an extra element as part of the secret key. Due to this reason (and also because of the use of symmetric authentication) the efficiency of encryption and decryption is improved. This, however, comes at a cost. For one thing, the underlying assumption is stronger; secondly, the number of group elements in the private key is more than that used in the current scheme and the KG-IBE. Also, the key generation time is more, though this is of less significance, since key generation is a less frequent activity.

For the KG and KV schemes, Table 6 shows the costs assuming symmetric pairings. For the same security level, the sizes of representations of group elements for symmetric pairing implementation is a few times larger than that for asymmetric pairing implementation. The KG schemes can be converted to the setting of asymmetric pairings. This, however, will not be useful, since the decryption algorithm (with or without using implicit rejection) will be slower than the decryption algorithms of IBE-1 and IBE-2. For KG with implicit rejection, basically, the time for the four extra scalar multiplications plus a hash of the identity will be more than the time difference arising between $2[P_2]$ and $1[P_3]$. This is due to the fact that we use symmetric key authentication to replace pairing operations, something which is not done by KG. Conversion of the KV scheme to the setting of asymmetric pairing (as has been considered in the full version) will provide faster encryption and decryption algorithms. The trade-off is that a stronger hardness assumption is used and also the private key size will be longer and key generation time will be more. In summary, if one is interested in IBE schemes based on the hardness of the DBDH assumption, then IBE-1 and IBE-2 are the currently known most efficient CCA-secure IBE schemes which are secure in the full model without the random oracle heuristic.

Hierarchical identity-based encryption. Based on the work by BMW (Boneh et al., 2005b), the KG paper (Kiltz and Galindo, 2009) sketches a construction of a HIBE scheme (in the setting of symmetric pairing). This is based on Waters (2005) HIBE scheme and so the number of public parameters is significantly more compared to the schemes described here. Also, the advantage of using symmetric key authentication to replace pairing computation extends to our HIBE schemes. As a result, for small depth $h = 2, 3$, the HIBE schemes described here are the best known schemes which are secure in the full model, does not use the random oracle heuristics and are based on the hardness of the DBDH problem.

7 Concluding remarks

In this paper, we revisited the problem of constructing practical (hierarchical) IBE based on the DBDH assumption. Our starting point is Waters (2005) (H)IBE scheme and its generalisations and improvements that had been proposed in our earlier conference papers (Chatterjee and Sarkar, 2005, 2006a). These (H)IBE constructions are recast in the most efficient setting of asymmetric pairings which is the Type 3 setting. Moving from symmetric to asymmetric pairing settings leads to several variants of the basic scheme with associated trade-offs.

We described two such variants of the CPA-secure HIBE and then showed how to modify them to obtain CCA-secure hybrid schemes. The final schemes are secure against adaptive adversaries (making both key extraction and decryption queries) without using the random oracle heuristic. Security is based on the hardness of the DBDH problem. To the best of our knowledge, in this setting, the IBE schemes described in this paper are the currently known most efficient constructions.

Following Naor's transformation, we have described how to obtain a HIBS scheme by modifying the CPA-secure HIBE schemes. Instances of the HIBS schemes result in a usual signature scheme as well as an IBS scheme both of which improve upon previously known proposals under the same assumption. Improvements of constructions for other cryptographic primitives are possible using the ideas given here. We mention two such examples.

Table 6 Comparison of parameter sizes and costs of different operations

Scheme	*assump*	*Parameter sizes*				*Times*		
		PP	*msk*	*pvt key*	*cpr txt*	*key gen*	*enc*	*dec*
IBE-1	DBDH	$(l+2,1,1,0)$	$(1,0,0,0)$	$(2,1,0,0)$	$(2,1,0,0)$	$1\left[\mathrm{H}^{(1)}_{n,l}\right] + 1[\mathrm{SM}_1] + 1[\mathrm{SM}_2]$	$1\left[\mathrm{H}^{(1)}_{n,l}\right] + 1[\mathrm{E}] + 3[\mathrm{SM}_1] + 1[\mathrm{SM}_2]$	$2[\mathrm{P}_2] + 1[\mathrm{SM}_1]$
IBE-2	DBDH	$(l+4,0,1,0)$	$(1,0,0,l+1)$	$(0,4,0,0)$	$(3,0,0,0)$	$2[\mathrm{SM}_2]$	$1\left[\mathrm{H}^{(1)}_{n,l}\right] + 1[\mathrm{E}] + 4[\mathrm{SM}_1]$	$2[\mathrm{P}_2] + 1[\mathrm{SM}_2]$
KG (Kiltz and Galindo, 2009)	DBDH	$((l+3,1,0))$	$((1,0,0))$	$((2,0,0))$	$((3,0,0))$	$1[\mathrm{H}_{n,l}] + 2[\mathrm{SM}]$	$1[\mathrm{H}_{n,l}] + 4[\mathrm{SM}] + 1[\mathrm{SE}]$	$3[\mathrm{SP}_2] + 1[\mathrm{H}_{n,l}] + 1[\mathrm{SM}]$
KG (Kiltz and Galindo, 2009) (imp rej)	DBDH	$((l+3,1,0))$	$((1,0,0))$	$((2,0,0))$	$((3,0,0))$	$1[\mathrm{H}_{n,l}] + 2[\mathrm{SM}]$	$1[\mathrm{H}_{n,l}] + 4[\mathrm{SM}] + 1[\mathrm{SE}]$	$1[\mathrm{SP}_3] + 1[\mathrm{H}_{n,l}] + 5[\mathrm{SM}]$
KV (Kiltz and Vahlis, 2008)	mBDDH	$((l+2,1,0))$	$((1,0,0))$	$((3,0,0))$	$((2,0,0))$	$1[\mathrm{H}_{n,l}] + 3[\mathrm{SM}]$	$1[\mathrm{H}_{n,l}] + 3[\mathrm{SM}] + 1[\mathrm{SE}]$	$1[\mathrm{SP}_2] + 1[\mathrm{SM}]$

Note: Costs of symmetric key operations are not shown.

1 A wildcard IBE (WIBE) extends the notion of HIBE. This primitive was introduced in Abdalla et al. (2006). Birkett et al. (2007) provided a construction of a CCA-secure WIBE by modifying the Kiltz-Galindo scheme (KG-HIBE) in Kiltz and Galindo (2009). The conversion from KG-HIBE to WIBE described in Birkett et al. (2007) can be applied to the HIBE described in this work to obtain a different WIBE.

   a As in Kiltz and Galindo (2009), the WIBE scheme in Birkett et al. (2007) works with symmetric pairings and requires $((h(n+1), 1, 1))$ size public parameters. Using our technique in the asymmetric pairing setting, this will come down to $(h+l, 1, 1, 0)$ where one can choose a suitable l between 1 and n. Even for $l = n$, the number of public parameters is substantially less than that of Birkett et al. (2007). This reduction in public parameters is inherited from the reduction in public parameters of the HIBE in Section 3.1 over the HIBE suggested in Waters (2005) and Kiltz and Galindo (2009).

   b Recall that the KG-HIBE makes several pairing-based verifications to check the well-formedness of the ciphertext. The CCA-secure HIBE scheme given in this work removes these pairings and performs well-formedness check using symmetric authentication. This results in a WIBE whose decryption algorithm is more efficient than what has been reported in Birkett et al. (2007). The details of the construction and the proof are fairly straightforward from the description given in Birkett et al. (2007) and the current work.

2 A 2-level Waters (2005) HIBE has been used to construct a CCA-secure certificateless encryption (CLE) scheme in Dent et al. (2008) using the setting of symmetric pairings. Again using the HIBE scheme of the present work, one obtains a CLE scheme in the setting of asymmetric pairings with significantly smaller size public parameters as well as a more efficient decryption algorithm.

Acknowledgements

We thank the anonymous reviewers for their helpful comments.

References

Abdalla, M., Catalano, D., Dent, A.W., Malone-Lee, J., Neven, G. and Smart, N.P. (2006) 'Identity-based encryption gone wild', in Bugliesi, M., Preneel, B., Sassone, V. and Wegener, I. (Eds.): *ICALP (2), Lecture Notes in Computer Science*, Vol. 4052, pp.300–311, Springer.

Abe, M., Gennaro, R., Kurosawa, K. and Shoup, V. (2005) 'Tag-KEM/DEM: a new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM', in Cramer, R. (Ed.): *Advances in Cryptology – EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Lecture Notes in Computer Science*, Aarhus, Denmark, Springer, 22–26 May, Vol. 3494, pp.128–146.

Batten, L.M. and Safavi-Naini, R. (Eds.) (2006) *Information Security and Privacy, 11th Australasian Conference, ACISP 2006, Proceedings, Lecture Notes in Computer Science*, Springer, Melbourne, Australia, 3–5 July, Vol. 4058.

Bellare, M. and Ristenpart, T. (2009) 'Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme', in Joux, A. (Ed.): *EUROCRYPT, Lecture Notes in Computer Science*, Vol. 5479, pp.407–424, Springer.

Birkett, J., Dent, A.W., Neven, G. and Schuldt, J.C.N. (2007) 'Efficient chosen-ciphertext secure identity-based encryption with wildcards', in Pieprzyk, J., Ghodosi, H. and Dawson, E. (Eds.): *ACISP, Lecture Notes in Computer Science*, Vol. 4586, pp.274–292, Springer.

Boneh, D. and Boyen, X. (2004a) 'Efficient selective-ID secure identity-based encryption without random oracles', in Cachin, C. and Camenisch, J. (Eds.): *Advances in Cryptology – EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Lecture Notes in Computer Science*, Interlaken, Switzerland, Springer, 2–6 May, Vol. 3027, pp.223–238.

Boneh, D. and Boyen, X. (2004b) 'Secure identity based encryption without random oracles', in Franklin, M.K. (Eds.): *Advances in Cryptology – CRYPTO 2004, 24th Annual International Cryptology Conference, Proceedings, Lecture Notes in Computer Science*, Springer, Santa Barbara, California, USA, 15–19 August, Vol. 3152, pp.443–459.

Boneh, D. and Boyen, X. (2004) 'Short signatures without random oracles', Cachin, C. and Camenisch, J. (Eds.): *Advances in Cryptology – EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Lecture Notes in Computer Science*, Interlaken, Switzerland, Springer, 2–6 May, Vol. 3027, pp.56–73.

Boneh, D. and Franklin, M.K. (2003) 'Identity-based encryption from the weil pairing', *SIAM J. Comput.*, Vol. 32, No. 3, pp.586–615; earlier version appeared in the *Proceedings of CRYPTO 2001*.

Boneh, D., Boyen, X. and Goh, E-J. (2005a) 'Hierarchical identity based encryption with constant size ciphertext', in Cramer, R. (Ed.): *Advances in Cryptology – EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Lecture Notes in Computer Science*, Aarhus, Denmark, Springer, 22–26 May, Vol. 3494, pp.440–456; Full version available at Cryptology ePrint Archive; Report 2005/015.

Boyen, X., Mei, Q. and Waters, B. (2005b) 'Direct chosen ciphertext security from identity-based techniques', in Atluri, V., Meadows, C. and Juels, A. (Eds.): ACM *Conference on Computer and Communications Security*, ACM, pp.320–329.

Boneh, D., Lynn, B. and Shacham, H. (2001) 'Short signatures from the Weil pairing', in Boyd, C. (Ed.): *ASIACRYPT, Lecture Notes in Computer Science*, Springer, Vol. 2248, pp.514–532.

Boyen, X. (2008a) 'Flexible IBE and beyond in the commutative-blinding framework', in Joye, M. and Neven, G. (Eds.): *Identity-Based Cryptography, Volume 2 of Cryptology and Information Security Series*, IOS Press.

Boyen, X. (2008b) 'The uber-assumption family', in Galbraith, S.D. and Paterson, K.G. (Eds.): *Pairing, Lecture Notes in Computer Science*, Vol. 5209, pp.39–56, Springer.

Cachin, C. and Camenisch, J. (Eds.) (2004) *Advances in Cryptology – EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Lecture Notes in Computer Science*, Interlaken, Switzerland, Springer, 2–6 May, Vol. 3027.

Canetti, R., Halevi, S. and Katz, J. (2004) 'Chosen-ciphertext security from identity-based encryption', in Cachin, C. and Camenisch, J. (Eds.) (2004) *Advances in Cryptology – EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Lecture Notes in Computer Science*, Interlaken, Switzerland, Springer, 2–6 May, Vol. 3027, pp.207–222.

Chatterjee, S. and Menezes, A. (2011) 'On cryptographic protocols employing asymmetric pairings – the role of Ψ revisited', *Discrete Applied Mathematics*, Vol. 159, No. 13, pp.1311–1322.

Chatterjee, S. and Sarkar, P. (2005) 'Trading time for space: towards an efficient IBE scheme with short(er) public parameters in the standard model', in Won, D.H. and Kim, S. (Eds.): *ICISC, Lecture Notes in Computer Science*, Springer, Vol. 3935, pp.424–440.

Chatterjee, S. and Sarkar, P. (2006a) 'HIBE with short public parameters without random oracle', in Lai, X. and Chen, K. (Eds.): *ASIACRYPT, Lecture Notes in Computer Science*, Springer, Vol. 4284, pp.145–160; see also Cryptology ePrint Archive, Report 2006/279.

Chatterjee, S. and Sarkar, P. (2006b) 'New constructions of constant size ciphertext HIBE without random Oracle', in Rhee, M.S. and Lee, B. (Eds.): *ICISC, Lecture Notes in Computer Science*, Springer, Vol. 4296, pp.310–327.

Chatterjee, S., Hankerson, D., Knapp, E. and Menezes, A. (2010) 'Comparing two pairing-based aggregate signature schemes', *Des. Codes Cryptography*, Vol. 55, Nos. 2–3, pp.141–167.

Chaum, D., Evertse, J-H. and van de Graaf, J. (1987) 'An improved protocol for demonstrating possession of discrete logarithms and some generalizations', in *EUROCRYPT*, pp.127–141.

Cramer, R. (Ed.) (2005) *Advances in Cryptology – EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Lecture Notes in Computer Science*, Aarhus, Denmark, Springer, 22–26 May, Vol. 3494.

Cramer, R. and Shoup, V. (2003) 'Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack', *SIAM J. Comput.*, Vol. 33, No. 1, pp.167–226.

Dent, A.W., Libert, B. and Paterson, K.G. (2008) 'Certificateless encryption schemes strongly secure in the standard model', in Cramer, R. (Ed.): *Public Key Cryptography, Lecture Notes in Computer Science*, Springer, Vol. 4939, pp.344–359.

Franklin, M.K. (Ed.) (2004) *Advances in Cryptology – CRYPTO 2004, 24th Annual International Cryptology Conference, Proceedings, Lecture Notes in Computer Science*, Springer, Santa Barbara, California, USA, 15–19 August, Vol. 3152.

Freeman, D., Scott, M. and Teske, E. (2010) 'A taxonomy of pairing-friendly elliptic curves', *J. Cryptology*, Vol. 23, No. 2, pp.224–280.

Galbraith, S.D., Paterson, K.G. and Smart, N.P. (2008) 'Pairings for cryptographers', *Discrete Applied Mathematics*, Vol. 156, No. 16, pp.3113–3121.

Galindo, D. (2004) 'The exact security of pairing based encryption and signature schemes', Based on a talk at *Workshop on Provable Security*, INRIA, Paris, available from author's website.

Gentry, C. (2006) 'Practical identity-based encryption without random oracles', in Vaudenay, S. (Ed.): *EUROCRYPT, Lecture Notes in Computer Science*, Springer, Vol. 4004, pp.445–464.

Gentry, C. and Halevi, S. (2009) 'Hierarchical identity based encryption with polynomially many levels', in Reingold, O. (Ed.): *TCC, Lecture Notes in Computer Science*, Springer, Vol. 5444, pp.437–456.

Gentry, C. and Silverberg, A. (2002) 'Hierarchical ID-based cryptography', in Zheng, Y. (Ed.): *ASIACRYPT, Lecture Notes in Computer Science*, Springer, Vol. 2501, pp.548–566.

Gentry, C., Peikert, C. and Vaikuntanathan, V. (2008) 'Trapdoors for hard lattices and new cryptographic constructions', in Ladner, R.E. and Dwork, C. (Eds.): *STOC, ACM*, pp.197–206.

Hess, F., Smart, N.P. and Vercauteren, F. (2006) 'The eta pairing revisited', *IEEE Transactions on Information Theory*, Vol. 52, No. 10, pp.4595–4602.

Hofheinz, D. and Kiltz, E. (2008) 'Programmable hash functions and their applications', in Wagner, D. (Ed.): *CRYPTO, Lecture Notes in Computer Science*, Springer, Vol. 5157, pp.21–38.

Horwitz, J. and Lynn, B. (2002) 'Toward hierarchical identity-based encryption', in Knudsen, L.R. (Ed.): *EUROCRYPT, Lecture Notes in Computer Science*, Springer, Vol. 2332, pp.466–481.

Kiltz, E. and Galindo, D. (2006) 'Direct chosen-ciphertext secure identity-based key encapsulation without random oracles', in Batten, L.M. and Safavi-Naini, R. (Eds.): *Information Security and Privacy, 11th Australasian Conference, ACISP 2006, Proceedings, Lecture Notes in Computer Science*, Springer, Melbourne, Australia, 3–5 July, Vol. 4058, pp.336–347.

Kiltz, E. and Galindo, D. (2009) 'Direct chosen-ciphertext secure identity-based key encapsulation without random oracles', *Theor. Comput. Sci.*, Vol. 410, Nos. 47–49, pp.5093–5111; Earlier version appeared as Kiltz and Galindo 2006.

Kiltz, E. and Vahlis, Y. (2008) 'CCA2 secure IBE: standard model efficiency through authenticated symmetric encryption', in Malkin, T. (Ed.): *CT-RSA, Lecture Notes in Computer Science*, Springer, Vol. 4964, pp.221–238.

Kurosawa, K. and Desmedt, Y. (2004) 'A new paradigm of hybrid encryption scheme', in Franklin, M.K. (Ed.): *Advances in Cryptology – CRYPTO 2004, 24th Annual International Cryptology Conference, Proceedings, Lecture Notes in Computer Science*, Springer, Santa Barbara, California, USA, 15–19 August, Vol. 3152, pp.426–442.

Lee, E., Lee, H-S. and Park, C-M. (2009) 'Efficient and generalized pairing computation on abelian varieties', *IEEE Transactions on Information Theory*, Vol. 55, No. 4, pp.1793–1803.

Lenstra, A.K. and Verheul, E.R. (2001) 'Selecting cryptographic key sizes', J. Cryptology, Vol. 14, No. 4, pp.255–293.

Lewko, A.B. (2012) 'Tools for simulating features of composite order bilinear groups in the prime order setting', in Pointcheval, D. and Johansson, T. (Eds.): *EUROCRYPT, Lecture Notes in Computer Science*, Springer, Vol. 7237, pp.318–335.

Lewko, A.B. and Waters, B. (2010) 'New techniques for dual system encryption and fully secure HIBE with short ciphertexts', in Micciancio, D. (Ed.): TCC, *Lecture Notes in Computer Science*, Springer, Vol. 5978, pp.455–479.

Lewko, A.B. and Waters, B. (2011) 'Unbounded HIBE and attribute-based encryption', in Paterson, K.G. (Ed.): *EUROCRYPT, Lecture Notes in Computer Science*, Springer, Vol. 6632, pp.547–567.

Motwani, R. and Raghavan, P. (1995) *Randomized Algorithms*, Cambridge University Press, USA.

Naccache, D. (2007) 'Secure and practical identity-based encryption', *IET Information Security*, Vol. 1, No. 2, pp.59–64.

Paterson, K.G. and Schuldt, J.C.N. (2006) 'Efficient identity-based signatures secure in the standard model', in Batten, L.M. and Safavi-Naini, R. (Eds.): *Information Security and Privacy, 11th Australasian Conference, ACISP 2006, Proceedings, Lecture Notes in Computer Science*, Springer, Melbourne, Australia, 3–5 July, Vol. 4058, pp.207–222.

Ramanna, S.C., Chatterjee, S. and Sarkar, P. (2012) 'Variants of Waters' dual system primitives using asymmetric pairings – (extended abstract)', in Fischlin, M., Buchmann, J. and Manulis, M. (Eds.): *Public Key Cryptography, Lecture Notes in Computer Science*, Springer, Vol. 7293, pp.298–315.

Rogaway, P. (2004) 'Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC', in Lee, P.J. (Ed.): *ASIACRYPT, Lecture Notes in Computer Science*, Springer, Vol. 3329, pp.16–31.

Rogaway, P. (2006) 'Formalizing human ignorance', in Nguyen, P.Q. (Ed.): *VIETCRYPT, Lecture Notes in Computer Science*, Springer, Vol. 4341, pp.211–228.

Sarkar, P. (2010) 'Pseudo-random functions and parallelizable modes of operations of a block cipher', *IEEE Transactions on Information Theory*, Vol. 56, No. 8, pp.4025–4037.

Sarkar, P. and Chatterjee, S. (2007) 'Construction of a hybrid hibe protocol secure against adaptive attacks', in Susilo, W., Liu, J.K. and Mu, Y. (Eds.): *ProvSec, Lecture Notes in Computer Science*, Springer, Vol. 4784, pp.51–67.

Shamir, A. (1984) 'Identity-based cryptosystems and signature schemes', in Blakley, G.R. and Chaum, D. (Eds.): *CRYPTO, Lecture Notes in Computer Science*, Springer, Vol. 196, pp.47–53.

Shi, E. and Waters, B. (2008) 'Delegating capabilities in predicate encryption systems', in Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A. and Walukiewicz, I. (Eds.): *ICALP (2), Lecture Notes in Computer Science*, Springer, Vol. 5126, pp.560–578.

Shoup, V. (2004) 'Sequences of games: a tool for taming complexity in security proofs', *Cryptology ePrint Archive*, Report 2004/332.

Smart, N.P. and Vercauteren, F. (2007) 'On computable isomorphisms in efficient asymmetric pairing-based systems', *Discrete Applied Mathematics*, Vol. 155, No. 4, pp.538–547.

Vercauteren, F. (2010) 'Optimal pairings', *IEEE Transactions on Information Theory*, Vol. 56, No. 1, pp.455–461.

Waters, B. (2005) 'Efficient identity-based encryption without random oracles', in Cramer, R. (Ed.): *Advances in Cryptology – EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Lecture Notes in Computer Science*, Aarhus, Denmark, Springer, 22–26 May, Vol. 3494, pp.114–127.

Waters, B. (2009) 'Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions', in Halevi, S. (Ed.): *CRYPTO, Lecture Notes in Computer Science*, Springer, Vol. 5677, pp.619–636.

Notes

1 Parts of the paper have appeared in different and abridged forms in Chatterjee and Sarkar (2005, 2006a) and Sarkar and Chatterjee (2007).

2 Including the versions of the schemes in our conference papers upon which this work is based.

3 We remark that a similar reduction of the computational cost in key generation is possible in the IBE version of HIBE-1. However, there is no associated reduction in the size of the public parameter as U_1' and the U_is are required for the encryption also.