Generalization of the Selective-ID Security Model for HIBE Protocols

Sanjit Chatterjee and Palash Sarkar

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

Abstract. We generalize the selective-ID security model for HIBE by introducing two new security models. Broadly speaking, both these models allow the adversary to commit to a set of identities and in the challenge phase choose any one of the previously committed identities. Two constructions of HIBE are presented which are secure in the two models. Further, we show that the HIBEs can be modified to obtain a multiple receiver IBE which is secure in the selective-ID model without the random oracle assumption.

1 Introduction

Identity based encryption (IBE) was introduced by Shamir [17]. This is a public key encryption protocol where the public key can be any string. The corresponding private key is generated by a private key generator (PKG) and provided to the user in an offline phase. The notion of IBE can simplify many applications of public key encryption (PKE) and is currently an active research area.

The notion of the IBE was later extended to hierarchical IBE (HIBE) [15, 16]. In an IBE, the PKG has to generate the private key for any identity. The notion of the HIBE reduces the workload of the PKG by delegating the private key generation task to lower level entities, i.e., entities who have already obtained their private keys. Though a HIBE by itself is an interesting cryptographic primitive, it can also be used to construct other primitives like forward secure encryption and broadcast encryption protocols.

The first efficient construction of an IBE was provided by Boneh and Franklin [7]. This paper also introduced an appropriate security model for IBE. The proof of security in [7] used the so-called random oracle assumption. This started a search for constructions which can be proved to be secure without the random oracle assumption. The first such construction of an IBE was given in [10]. However, the IBE in [10] can only be proved to be secure in a weaker model (the selective-ID model) as opposed to the full model considered in [7]. Later Boneh and Boyen [3] presented a more efficient construction of HIBE which is secure in the selective-ID (sID) model without the random oracle assumption.

The full security model in [7] allows an adversary to adaptively ask the PKG for private keys of identities of its choosing. (The security model also allows decryption queries, which we ignore for the present.) Then it submits two messages M_0, M_1 and an identity v^* and is given an encryption of M_γ under v^*, where γ is a randomly chosen bit. The identity v^* can be any identity other than those for which the adversary has already obtained the private key or can easily obtain the private key from the information it has received. The main difficulty in obtaining an efficient construction of a HIBE which is secure in this model is the wide flexibility of the adversary in choosing v^*.

The sID model attempts to curb the adversary's flexibility in the following manner. In the game between the adversary and the simulator, the adversary has to commit to an identity even before the HIBE protocol is set-up by the simulator. The simulator then sets up the HIBE. This allows the simulator to set-up the HIBE based on the identity committed by the adversary. In the actual game, the adversary cannot ask for the private key of the committed identity (or of any of its prefixes, in the case of HIBE). During the challenge stage, the adversary submits two messages M_0, M_1 as usual and is given an encryption of M_γ under the previously fixed identity v^*. Note that this is significantly more restrictive than the full model since the adversary has to commit to an identity even before it sees the public parameters of the HIBE.

Our Contributions: In this paper, we generalize the sID model and introduce two new models of security for HIBE protocols. The basic idea is to modify the security game so as to allow the adversary to commit to a set of identities (instead of one identity in the sID model) before set-up. During the game, the adversary can execute key extraction queries on any identity not in the committed set. In the challenge stage, the challenge identity is chosen by the adversary from among the set that it has previously committed to.

For IBE, this is a strict generalization of the sID model, since we can get the sID model by enforcing the size of the committed set of identities to be one. On the other hand, for HIBE, there are two ways to view this generalization leading to two different security models $\mathcal{M}_1$ and $\mathcal{M}_2$.

In $\mathcal{M}_1$, the adversary commits to a set $\mathcal{I}^*$. It can then ask for the private key of any identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$ as long as all the v_is are not in $\mathcal{I}^*$. Further, during the challenge stage, it has to submit an identity all of whose components are in $\mathcal{I}^*$. If we restrict the adversary to only single component identities (i.e., we are considering only the IBE protocols), then this is a clear generalization of the sID model for IBE. On the other hand, in the case of HIBE, we cannot fix the parameters of this model to obtain the sID model for HIBE.

The second model, $\mathcal{M}_2$, is an obvious generalization of the sID model for HIBE. In this case, the adversary specifies τ sets $\mathcal{I}_1^*, \ldots, \mathcal{I}_\tau^*$. Then it can ask for private key of any identity v as long as there is an i such that the ith component of v is not in $\mathcal{I}_i^*$. In the challenge stage, the adversary has to submit an identity such that for all i, the ith component of the identity is in $\mathcal{I}_i^*$.

Even though $\mathcal{M}_2$ generalizes the sID model for HIBE, we think $\mathcal{M}_1$ is also an appropriate model for a HIBE protocol. The adversary would be specifying a set of "sensitive" keywords to be $\mathcal{I}^*$. It can then ask for the private key of any identity as long as one component of the identity is not sensitive and in the challenge stage has to submit an identity all of whose components are sensitive. The added flexibility in $\mathcal{M}_2$ is that the adversary can specify different sets of sensitive keywords for the different levels of HIBE. In practice, this flexibility might not be required since keywords like root, admin, dba, etcetera will be sensitive for all levels.

We present two constructions of HIBE denoted by $\mathcal{H}_1$ and $\mathcal{H}_2$. $\mathcal{H}_1$ is proved to be secure in the model $\mathcal{M}_1$ under the DBDH assumption while $\mathcal{H}_2$ is proved to be secure in the model $\mathcal{M}_2$ also under the DBDH assumption. Our constructions and proofs of security are very similar to that of the Boneh-Boyen HIBE (BB-HIBE) [3]. The actual technical novelty in the proofs is the use of a polynomial, which in the case of the BB-HIBE is of degree one. The use of an appropriate polynomial of degree greater than one allows us to prove security in the more general models $\mathcal{M}_1$ and $\mathcal{M}_2$. However, this flexibility comes at a cost. In both $\mathcal{H}_1$ and $\mathcal{H}_2$, the number of required scalar multiplications increases linearly with the size of the committed set of identities.

Multiple receiver IBE (MR-IBE) is an interesting concept which was introduced by Baek, Safavi-Naini and Susilo [1]. In an MR-IBE, an encryptor can encrypt a message in such a way that any one of a set of identities can decrypt the message. A trivial way to achieve this is to separately encrypt the message several times. It turns out that the efficiency can be improved. A more efficient construction of MR-IBE was presented in [1]. The proof of security was in the sID model under the *random oracle* assumption.

We show that the HIBE $\mathcal{H}_1$ or $\mathcal{H}_2$ when restricted to IBE can be easily modified to obtain an efficient MR-IBE. Our MR-IBE is proved to be secure in the sID model *without* the random oracle assumption and to the best of our knowledge this is the first of such kind.

2 Preliminaries

2.1 Cryptographic Bilinear Map

Let G_1 and G_2 be cyclic groups of same prime order p and $G_1 = \langle P \rangle$, where we write G_1 additively and G_2 multiplicatively. A mapping $e : G_1 \times G_1 \to G_2$ is called a cryptographic bilinear map if it satisfies the following properties:

- Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in \mathbb{Z}_p$.
- Non-degeneracy: If $G_1 = \langle P \rangle$, then $G_2 = \langle e(P, P) \rangle$.
- Computability: There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

Since $e(aP, bP) = e(P, P)^{ab} = e(bP, aP)$, $e()$ also satisfies the symmetry property. Weil pairing [6] and Tate pairing [2, 14] are examples of cryptographic bilinear maps. General definitions of Weil and Tate pairings do not satisfy the symmetry property but for certain curves they can be modified to satisfy the symmetry property. In this paper, we will use symmetric bilinear maps.

The only known examples of $e()$ have G_1 to be a group of Elliptic Curve (EC) points and G_2 to be a subgroup of a multiplicative group of a finite field. Hence, in papers on pairing implementations [2, 14], it is customary to write G_1 additively and G_2 multiplicatively. On the other hand, some "pure" protocol papers [3, 4, 18] write both G_1 and G_2 multiplicatively, though this is not true of the early protocol papers [6, 15]. Here we follow the first convention as it is closer to the known examples.

2.2 Hardness Assumption

Decision Bilinear Diffie-Hellman (DBDH) Problem: The DBDH problem in $\langle G_1, G_2, e() \rangle$ [7] is as follows: Given a tuple $\langle P, aP, bP, cP, Z \rangle$, where $Z \in G_2$, decide whether $Z = e(P, P)^{abc}$ (which we denote as Z $\mathsf{is\ real}$) or whether Z is a random element of G_2 (which we denote as Z $\mathsf{is\ random}$). Let $\mathcal{B}$ be a probabilistic algorithm which takes as input a tuple $\langle P, aP, bP, cP, Z \rangle$ and outputs a bit. The advantage of $\mathcal{B}$ in solving the DBDH problem is defined to be

$$\begin{aligned}\mathsf{Adv}_{\mathcal{B}}^{\mathsf{DBDH}} = |&\mathsf{Pr}[\mathcal{B}(P, aP, bP, cP, Z) = 1 | \mathsf{Z\ is\ real}] \\ &-\mathsf{Pr}[\mathcal{B}(P, aP, bP, cP, Z) = 1 |\ \mathsf{Z\ is\ random}]|\end{aligned}$$

where the probabilities are 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 $\mathsf{Adv}^{\mathsf{DBDH}}(t)$ denotes the maximum of $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{DBDH}}$ where the maximum is taken over all adversaries running in time at most t.

2.3 HIBE Protocol

Following [16, 15] a hierarchical identity based encryption (HIBE) scheme is specified by four probabilistic algorithms: Setup, Key Generation, Encryption and Decryption.

Setup: It takes as input a security parameter and returns the system parameters together with the master key. The system parameters are publicly known while the master key is known only to the private key generator (PKG).

The system parameters include a description of the message space, the ciphertext space and the identity space. An identity of depth τ is a tuple $(\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$, where each v_j is an element of a set $\mathcal{I}$. From an application point of view, we would like $\mathcal{I}$ to be the set of all binary strings. On the other hand, for construction purposes, this is too general and one usually requires $\mathcal{I}$ to have an algebraic structure. The two requirements are met by assuming that a collision resistant hash function maps an arbitrary string to the set $\mathcal{I}$ having an algebraic structure.

A special case of a HIBE protocol arises when only single component identities are allowed. In this case, the protocol is said to be simply an identity based encryption (IBE) protocol.

Key Generation: The task of this algorithm is to assign a private key d_v for an identity v of depth τ. To this end, it takes as input an identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$ of depth τ and the private key $d_{|\tau-1}$ corresponding to the identity $\mathsf{v}_{|\tau-1} = (\mathsf{v}_1, \ldots, \mathsf{v}_{\tau-1})$ and returns d_v. In the case $\tau = 1$, the private key $d_{|\tau-1}$ is the master key of the PKG and the key generation is done by the PKG. In the case $\tau > 1$, the private key corresponding to $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$ can be generated by the entity whose identity is $\mathsf{v}_{|\tau-1} = (\mathsf{v}_1, \ldots, \mathsf{v}_{\tau-1})$ and who has already obtained his/her private key $d_{|\tau-1}$.

Encryption: The encryption algorithm takes as input the identity v, the public parameters of the PKG and a message from the message space and produces a ciphertext in the cipher space.

Decryption: The decryption algorithm takes as input the ciphertext, the identity v under which encryption has been performed, the private key d_v of the corresponding identity v and the public parameters. It returns the message or bad if the ciphertext is not valid.

2.4 BB-HIBE

Set-Up: Let G_1, G_2 and $e()$ be as defined in Section 2. Let h be a positive integer which specifies the maximum depth of the HIBE. The identity space consists of all tuples $(\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$, $\tau \leq h$, where each $\mathsf{v}_i \in \mathbb{Z}_p$. The message space is G_2. The ciphertext corresponding to an identity $(\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$ is a tuple $(A, B, C_1, \ldots, C_\tau)$, where $A \in G_2$ and $B, C_1, \ldots, C_\tau \in G_1$.

Randomly choose $\alpha \in \mathbb{Z}_p$ and set $P_1 = \alpha P$. Choose P_2, $P_{3,1}, \ldots, P_{3,h}$ randomly from G_1. The public parameters are

$$(P, P_1, P_2, P_{3,1}, \ldots, P_{3,h})$$

and the master secret key is αP_2. The parameters P_1 and P_2 are not directly required in either encryption or decryption. We may replace them in the public parameters by $e(P_1, P_2)$. This will save the pairing computation during encryption.

A Notation: For any $y \in Z_p$, define $V_i(y) = P_{3,i} + yP_2$. We will simply write V_i for $V_i(\mathsf{v}_i)$.

Key Generation: Let $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$, $1 \leq \tau \leq h$ be an identity. The private key d_v corresponding to v is defined to be

$$(\alpha P_2 + r_1 V_1 + \ldots + r_\tau V_\tau, r_1 P, \ldots, r_\tau P)$$

where $r_1, \ldots, r_\tau$ are random elements of $\mathbb{Z}_p$. It can be shown [3] that the knowledge of a random private key corresponding to the tuple $(\mathsf{v}_1, \ldots, \mathsf{v}_{\tau-1})$ allows the generation of a random private key corresponding to v.

Encryption: Suppose a message M is to be encrypted under the identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$. Choose a random $t \in Z_p$. The ciphertext is $(A, B, C_1, \ldots, C_\tau)$, where

$$A = M \times e(P_1, P_2)^t; \quad B = tP; \quad C_i = tV_i, \text{ for } 1 \leq i \leq \tau.$$

Decryption: Suppose $(A, B, C_1, \ldots, C_\tau)$ is to be decrypted using the private key $(d_0, d_1, \ldots, d_\tau)$ corresponding to the identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$. Compute

$$A \times \frac{\prod_{i=1}^{\tau} e(d_i, C_i)}{e(d_0, B)}.$$

Again, it is standard to verify that the above computation yields M.

3 Security Model for HIBE

3.1 Security Model

The security model for HIBE is defined as an interactive game between an adversary and a simulator. Both the adversary and the simulator are modeled as probabilistic algorithms. Currently, there are two security models for HIBE – the selective-ID (sID) model and the full model. We will be interested in defining two new security models. We present the description of the interactive game in a manner which will help in obtaining a unified view of the sID, full and the new security models that we define.

In the game, the adversary is allowed to query two oracles – a decryption oracle $\mathcal{O}_d$ and a key-extraction oracle $\mathcal{O}_k$. The game has several stages.

Adversary's Commitment: In this stage, the adversary commits to two sets $\mathcal{S}_1$ and $\mathcal{S}_2$ of identities. The commitment has the following two consequences.

1. The adversary is not allowed to query $\mathcal{O}_k$ on any identity in $\mathcal{S}_1$ or on a prefix of any identity in $\mathcal{S}_1$.
2. In the challenge stage, the adversary has to choose one of the identities from the set $\mathcal{S}_2$.

There is a technical difficulty here. Note that the adversary has to commit to a set of identities even before the HIBE protocol has been set-up. On the other hand, the identity space is specified by the set-up algorithm of the HIBE protocol. In effect, this means that the adversary has to commit to identities even before it knows the set of identities. Clearly, this is not possible.

One possible way out is to allow the adversary to commit to binary strings and later when the set-up program has been executed, these binary strings are mapped to identities using a collision resistant hash functions. Another solution is to run the set-up program in two phases. In the first phase, the identity space is specified and is made available to the adversary; then the adversary commits to $\mathcal{S}_1$ and $\mathcal{S}_2$; and after obtaining $\mathcal{S}_1$ and $\mathcal{S}_2$ the rest of the set-up program is executed.

The above two approaches are not necessarily equivalent and may have different security consequences. On the other hand, note that if $\mathcal{S}_1 = \emptyset$ and $\mathcal{S}_2$ is the set of all identities (as is true in the full model), then this technical difficulty does not arise.

Set-Up: The simulator sets up the HIBE protocol and provides the public parameters to the adversary and keeps the master key to itself. Note that at this stage, the simulator knows $\mathcal{S}_1, \mathcal{S}_2$ and could possibly set-up the HIBE based on this knowledge. However, while doing this, the simulator must ensure that the probability distribution of the public parameters remains the same as in the specification of the actual HIBE protocol.

Phase 1: The adversary makes a finite number of queries where each query is addressed either to $\mathcal{O}_d$ or to $\mathcal{O}_k$. In a query to $\mathcal{O}_d$, it provides the ciphertext as well as the identity under which it wants the decryption. The simulator returns either the corresponding message or bad if the ciphertext is malformed. Similarly, in a query to $\mathcal{O}_k$, it asks for the private key of the identity it provides. This identity cannot be an element of $\mathcal{S}_1$ and neither can it be a prefix of any element in $\mathcal{S}_1$. Further, the adversary is allowed to make these queries adaptively, i.e., any query may depend on the previous queries as well as their answers.

Certain queries are useless and we will assume that the adversary does not make such queries. For example, if an adversary has queried $\mathcal{O}_k$ on any identity, then it is not allowed to present the same identity to $\mathcal{O}_d$ as part of a decryption query. The rationale is that since the adversary already has the private key, it can itself decrypt the required ciphertext.

Challenge: The adversary provides the simulator with an identity $\mathsf{v}^* \in \mathcal{S}_2$ and two messages M_0 and M_1. There is a restriction that the simulator should not have queried $\mathcal{O}_k$ for the private key of v^* or for the private key of any prefix of v^* in Phase 1. The simulator randomly chooses a $\gamma \in \{0,1\}$ and returns the encryption of M_γ under v^* to the adversary.

Phase 2: The adversary issues additional queries just as in Phase 1 with the following restrictions. It cannot ask $\mathcal{O}_d$ for the decryption of C^* under v^*; cannot ask $\mathcal{O}_k$ for the private key of any prefix of an identity in $\mathcal{S}_1$; and cannot make any useless query.

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

Adversary's Success: The adversary wins the game if it can successfully guess γ, i.e., if $\gamma = \gamma'$. The advantage of an adversary $\mathcal{A}$ in attacking the HIBE scheme is defined as:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HIBE}} = |\mathsf{Pr}[(\gamma = \gamma')] - 1/2|.$$

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 $\mathcal{O}_d$ and at most q_{ID} queries to $\mathcal{O}_k$.

A HIBE protocol is said to be $(\epsilon, t, q_{\mathsf{ID}}, q_{\mathsf{C}})$-CCA secure if $\mathsf{Adv}^{\mathsf{HIBE}}(t, q_{\mathsf{ID}}, q_{\mathsf{C}}) \leq \epsilon$. Any HIBE protocol secure against such an adversary is said to be secure against chosen ciphertext attack (CCA). A weaker version of security does not allow the adversary to make decryption queries, i.e., the adversary is not given access to $\mathcal{O}_d$. A HIBE protocol secure against such a weaker adversary is said to be secure against chosen plaintext attack (CPA). $\mathsf{Adv}^{\mathsf{HIBE}}(t, q)$ in this context 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 protocol is said to be (ϵ, t, q)-CPA secure if $\mathsf{Adv}^{\mathsf{HIBE}}(t, q) \leq \epsilon$. There are generic [10, 11] as well as non-generic methods [8] for converting a CPA-secure HIBE into a CCA-secure HIBE. Hence, in this paper, we will only consider construction of CPA-secure HIBE.

3.2 Full Model

Suppose $\mathcal{S}_1 = \emptyset$ and $\mathcal{S}_2$ is the set of all identities. By the rules of the game, the adversary is not allowed to query $\mathcal{O}_k$ on any identity in $\mathcal{S}_1$. Since $\mathcal{S}_1$ is empty, this means that the adversary is actually allowed to query $\mathcal{O}_k$ on any identity. Further, since $\mathcal{S}_2$ is the set of all identities, in the challenge stage, the adversary is allowed to choose any identity. In effect, this means that the adversary does not really commit to anything before set-up and hence, in this case, the commitment stage can be done away with. This particular choice of $\mathcal{S}_1$ and $\mathcal{S}_2$ is called the full model and is currently believed to be the most general notion of security for HIBE.

Note that the challenge stage restrictions as well as the restrictions in Phase 2 still apply.

3.3 Selective-ID Model

Let $\mathcal{S}_1 = \mathcal{S}_2$ be a singleton set. This means that the adversary commits to one particular identity; does not ask for a private key of any of its prefixes; and in the challenge phase is given the encryption of M_γ under this particular identity. This model is significantly weaker than the full model and is called the selective-ID model. The model was formally introduced in [11].

3.4 New Security Models

We introduce two new security models by suitably defining the sets $\mathcal{S}_1$ and $\mathcal{S}_2$. In our new models, (as well as the sID model), we have $\mathcal{S}_1 = \mathcal{S}_2$. (Note that in the full model, $\mathcal{S}_1 = \overline{\mathcal{S}_2}$.)

Model $\boldsymbol{\mathcal{M}}_1$*:* Let $\mathcal{I}^*$ be a set. Define $\mathcal{S}_1 = \mathcal{S}_2$ to be the set of all tuples $(\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$, such that each $\mathsf{v}_i \in \mathcal{I}^*$. If the HIBE is of maximum depth h, then $1 \leq \tau \leq h$. The value of τ is *not* fixed by the adversary in the commit phase.

Let us now see what this means. In the commit phase, the adversary commits to a set $\mathcal{I}^*$; never asks for a private key of an identity all of whose components are in $\mathcal{I}^*$; and during the challenge phase presents an identity all of whose components are in $\mathcal{I}^*$.

Consider the case of IBE, i.e., $h = 1$, which means that only single component identities are allowed. Then, we have $\mathcal{S}_1 = \mathcal{S}_2 = \mathcal{I}^*$. Let $|\mathcal{I}^*| = n$. If we put $n = 1$, then we obtain the sID model for IBE as discussed in Section 3.3. In other words, for IBE protocol, $\boldsymbol{\mathcal{M}}_1$ is a strict generalization of sID model.

If $h > 1$, then we have proper HIBE. In this case, $\boldsymbol{\mathcal{M}}_1$ differs fundamentally from the sID model.

1. In the sID model, the adversary is allowed to query $\mathcal{O}_k$ on a permutation of the challenge identity. This is not allowed in $\boldsymbol{\mathcal{M}}_1$.

2. In the sID model, the length of the challenge identity is fixed by the adversary in the commit phase. On the other hand, in $\boldsymbol{\mathcal{M}}_1$, the adversary is free to choose this length (to be between 1 and h) in the challenge stage itself.

In the case of HIBE, model $\boldsymbol{\mathcal{M}}_1$ is no longer a strict generalization of the usual sID model for HIBE. We cannot restrict the parameters of the model $\boldsymbol{\mathcal{M}}_1$ in any manner and obtain the sID model for HIBE. Thus, in this case, $\boldsymbol{\mathcal{M}}_1$ must be considered to be a new model.

Model $\boldsymbol{\mathcal{M}}_2$: Let $\mathcal{I}_1^*, \ldots, \mathcal{I}_\tau^*$ be sets and $|\mathcal{I}_j^*| = n_j$ for $1 \leq j \leq \tau$. We set

$$\mathcal{S}_1 = \mathcal{S}_2 = \mathcal{I}_1^* \times \cdots \times \mathcal{I}_\tau^*.$$

If the maximum depth of the HIBE is h, then $1 \leq \tau \leq h$.

In this model, for $1 \leq j \leq \tau$, the adversary is not allowed to obtain a private key for an identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_j)$ such that $\mathsf{v}_i \in \mathcal{I}_i^*$ for all $1 \leq i \leq j$. Further, the challenge identity is a tuple $(\mathsf{v}_1^*, \ldots, \mathsf{v}_\tau^*)$, with $\mathsf{v}_i \in \mathcal{I}_i^*$ for all $1 \leq i \leq \tau$. Like the sID model, the length of the challenge identity is fixed by the adversary in the commit phase.

This model is a strict generalization of the sID model for HIBE. This can be seen by setting $n_1 = \cdots = n_h = 1$, i.e., setting $\mathcal{I}_1^*, \ldots, \mathcal{I}_h^*$ to be singleton sets. On the other hand, $\boldsymbol{\mathcal{M}}_2$ and $\boldsymbol{\mathcal{M}}_1$ are not comparable due to at least two reasons.

1. In $\boldsymbol{\mathcal{M}}_1$, the length of the challenge identity can vary, while in $\boldsymbol{\mathcal{M}}_2$, the length is fixed in the commit phase.
2. In $\boldsymbol{\mathcal{M}}_2$, it may be possible for the adversary to obtain the private key for a permutation of the challenge identity, which is not allowed in $\boldsymbol{\mathcal{M}}_1$.

These two reasons are similar to the reasons for the difference between sID and $\boldsymbol{\mathcal{M}}_1$.

Parametrizing the models: A HIBE may have a bound on the maximum number of levels that can be supported. The corresponding security model also has the same restriction. For example, a HIBE of at most h levels will be called h-sID secure if it is secure in the sID model. The models $\boldsymbol{\mathcal{M}}_1$ and $\boldsymbol{\mathcal{M}}_2$ have additional parameters.

For the case of $\boldsymbol{\mathcal{M}}_1$, there is a single parameter n, which specifies the size of $\mathcal{I}^*$, the set which the adversary specifies in the commit phase. In this case, we will talk of (h,n)-$\boldsymbol{\mathcal{M}}_1$ security for an HIBE. Similarly, for $\boldsymbol{\mathcal{M}}_2$, we will talk of $(h, n_1, \ldots, n_h)$-$\boldsymbol{\mathcal{M}}_2$ security. Note that $(h, 1 \ldots, 1)$-$\boldsymbol{\mathcal{M}}_2$ model is same as the h-sID model.

4 Interpreting Security Models

The full security model is currently believed to provide the most general security model for HIBE. In other words, it provides any entity (having any particular identity) in the HIBE with the most satisfactory security assurance that the entity can hope for. The notion of security based on an appropriate adversarial game is adapted from the corresponding notion for public key encryption and the security assurance provided in that setting also applies to the HIBE setting. The additional consideration is that of identity and the key extraction queries to $\mathcal{O}_k$. We may consider the identity present during the challenge stage to be a target identity. In other words, the adversary wishes to break the security of the corresponding entity. In the full model, the target identity can be any identity, with the usual restriction that the adversary does not know the private key corresponding to this identity or one of its prefixes.

From the viewpoint of an individual entity e in the HIBE structure, the adversary's behavior appears to be the following. The adversary can possibly corrupt any entity in the structure, but as long as it is not able to corrupt that particular entity e or one of its ancestors, then it will not be able to succeed in an attack where the target identity is that of e. In other words, obtaining the private keys corresponding to the other identities does not help the adversary. Intuitively, that is the maximum protection that any entity e can expect from the system.

Let's reflect on the sID model. In this model, the adversary commits to an identity even before the set-up of the HIBE is done. The actual set-up can depend on the identity in question. Now consider the security assurance obtained by an individual entity e. Entity e can be convinced that if the adversary had targeted its identity and then the HIBE structure was set-up, in that case the adversary will not be successful in attacking it. Alternatively, e can be convinced that the HIBE structure can be set-up so as to protect it. Inherently, the sID model assures that the HIBE structure can be set-up to protect any identity, but only one.

Suppose that a HIBE structure which is secure in the sID model has already been set-up. It has possibly been set-up to protect one particular identity. The question now is what protection does it offer to entities with other identities? The model does not assure that other identities will be protected. Of course, this does not mean that other identities are vulnerable. The model simply does not say anything about these identities.

The system designer's point of view also needs to be considered. While setting up the HIBE structure, the designer needs to ensure security. The HIBE is known to be secure in the sID model and hence has a proof of security. The designer will play the role of the simulator in the security game. In the game, the adversary commits to an identity and then the HIBE is set-up so as to protect this identity. However, since the actual set-up has not been done, there is no real adversary and hence no real target identity. Thus, the designer has to assume that the adversary will probably be targeting some sensitive identity like root. The designer can then set-up the HIBE so as to protect this identity. However, once the HIBE has been set-up, the designer cannot say anything about the security of other possible sensitive identities like sysadmin. This is a limitation of the sID model.

It has been observed in [3] that a generic conversion from an IBE protocol secure in the selective-ID model to a protocol secure in the full model suffers from a security degradation by a factor of 2^ℓ, where identities are ℓ-bit strings. This also indicates the inadequacy of the selective-ID model.

This brings us to the generalization of the sID model that we have introduced. First consider the model $\mathcal{M}_1$ as it applies to IBE. In this model, the designer can assume that the adversary will possibly attack one out of a set of sensitive identities like $\{\mathsf{root}, \mathsf{admin}, \mathsf{dba}, \mathsf{sysadmin}\}$. It can then set-up the IBE so as to protect this set of identities. This offers better security than the sID model.

Now consider the model $\mathcal{M}_1$ as it applies to HIBE. In this case, the set $\mathcal{I}^*$ can be taken to be a set of sensitive keywords such as $\{\mathsf{root}, \mathsf{admin}, \mathsf{dba}, \mathsf{sysadmin}\}$. The adversary is not allowed to obtain private keys corresponding to identities all of whose components lie in $\mathcal{I}^*$. For the above example, the adversary cannot obtain the private key of (root, root), or (admin, root, dba). On the other hand, it is allowed to obtain keys corresponding to identities like (root, abracadabra). Thus, some of the components of the identities (on which key extraction query is made) may be in $\mathcal{I}^*$; as long as all of them are not in $\mathcal{I}^*$, the adversary can obtain the private key. On the other hand, all the components of the target identity have to be sensitive keywords, i.e., elements of $\mathcal{I}^*$. Clearly, model $\mathcal{M}_1$ provides an acceptable security notion for HIBE.

As mentioned earlier, a major difference of $\mathcal{M}_1$ with sID is that in sID the adversary is allowed to obtain a private key for a permutation of the challenge identity, whereas this is not allowed in $\mathcal{M}_1$. We point out that it is possible for a particular HIBE to be secure in both sID and $\mathcal{M}_1$. An example will be provided later. Thus, one may choose to obtain the good features of both sID and $\mathcal{M}_1$.

The model $\mathcal{M}_2$ is a clear generalization of the usual sID model for HIBE. The adversary fixes the sensitive keywords for each level of the HIBE up to the level it wishes to attack. It cannot make a key extraction query on an identity of depth τ, such that for $1 \leq i \leq \tau$, the ith component of the identity is among the pre-specified sensitive keywords for the ith level of the HIBE. Further, the target identity must be such that each of its component is a sensitive keyword for the corresponding HIBE level. As mentioned earlier, by fixing exactly one keyword for each level of the HIBE, we obtain the sID model.

The known protocols [4, 18, 12] which offer full model security suffer from security degradation. On the other hand, protocols such as [3, 5] which are secure in the selective-ID model have no security degradation. Thus, one can work with significantly smaller size groups while implementing the protocols in [3, 5] compared to the protocols in [4, 18, 12]. The protocols that are described in this paper have no security degradation. Hence, the group sizes used for implementing selective-ID protocols can be used for implementing the protocols secure in $\mathcal{M}_1$ and $\mathcal{M}_2$.

5 Constructions

We present several HIBE protocols which are proved to be secure in different models. In this section, we provide only the constructions. The security proofs are provided later.

The underlying groups G_1, G_2 and the pairing map $e(,)$ will be required by all the HIBE protocols. The set-up procedure of each HIBE will generate these groups based on the security parameter. The maximum depth of a HIBE will be denoted by h. In each of the HIBEs below, we will have P_1 and P_2 as public parameters which are not directly required. Instead, one may keep $e(P_1, P_2)$ in the public parameter which will save the pairing computation during encryption.

The components of identities are elements of $\mathbb{Z}_p$. Alternatively, if these are bit strings, then (as is standard) they will be hashed using a collision resistant hash function into $\mathbb{Z}_p$.

5.1 HIBE $\mathcal{H}_1$

Set-Up: The identity space consists of all tuples $(\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$, $\tau \leq h$, where each $\mathsf{v}_i \in \mathbb{Z}_p$. The message space is G_2. (In practical applications, the protocol will be converted into a hybrid encryption scheme where the message can be any binary string.) The ciphertext corresponding to an identity $(\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$ is a tuple $(A, B, C_1, \ldots, C_\tau)$, where $A \in G_2$ and $B, C_1, \ldots, C_\tau \in G_1$.

Randomly choose $\alpha \in \mathbb{Z}_p$ and set $P_1 = \alpha P$. Randomly choose P_2, $P_{3,1}, \ldots, P_{3,h}$, $Q_1, \ldots, Q_n$ from G_1 where n is a parameter. The public parameters are

$$(P, P_1, P_2, P_{3,1}, \ldots, P_{3,h}, Q_1, \ldots, Q_n)$$

and the master secret key is αP_2.

Notation: For any $y \in \mathbb{Z}_p$ define

$$V_i(y) = y^n Q_n + \cdots + y Q_1 + P_{3,i}.$$

Let $(\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$ be an identity. We write V_i for $V_i(\mathsf{v}_i)$.

Key Generation: The private key $d_{\mathsf{v}} = (d_0, d_1, \ldots, d_\tau)$ corresponding to an identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$ is defined to be

$$(d_0, d_1, \ldots, d_\tau) = (\alpha P_2 + r_1 V_1 + \ldots + r_\tau V_\tau, r_1 P, \ldots, r_\tau P)$$

where $r_1, \ldots, r_\tau$ are random elements of $\mathbb{Z}_p$. Key delegation can be done in the following manner. Let $(d'_0, d'_1, \ldots, d'_{\tau-1})$ be the private key corresponding to the identity $(\mathsf{v}_1, \ldots, \mathsf{v}_{\tau-1})$. Then $(d_0, d_1, \ldots, d_\tau)$ is obtained as follows. Choose a random r_τ from $\mathbb{Z}_p$ and define

$$\begin{aligned} d_0 &= d'_0 + r_\tau V_\tau; \\ d_i &= d'_i \qquad\qquad \text{for } 1 \le i \le \tau - 1; \\ d_\tau &= r_\tau P. \end{aligned}$$

This provides a proper private key corresponding to the identity $(\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$.

Encryption: Suppose a message M is to be encrypted under the identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$. Choose a random $t \in Z_p$. The ciphertext is $(A, B, C_1, \ldots, C_\tau)$, where

$$A = M \times e(P_1, P_2)^t; \quad B = tP; \quad C_i = tV_i, \text{ for } 1 \le i \le \tau.$$

Decryption: Suppose $(A, B, C_1, \ldots, C_\tau)$ is to be decrypted using the private key $(d_0, d_1, \ldots, d_\tau)$ corresponding to the identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$. Compute

$$\begin{aligned} A \times \frac{\prod_{i=1}^{\tau} e(d_i, C_i)}{e(d_0, B)} &= M \times e(P_1, P_2)^t \frac{\prod_{i=1}^{\tau} e(r_i P, tV_i)}{e(\alpha P_2 + \sum_{i=1}^{\tau} r_i V_i, tP)} \\ &= M \times e(P_1, P_2)^t \times \frac{1}{e(P_1, P_2)^t} \times \frac{\prod_{i=1}^{\tau} e(r_i P, tV_i)}{e(\sum_{i=1}^{\tau} r_i V_i, tP)} \\ &= M. \end{aligned}$$

Unbounded Depth HIBE: It is possible to modify $\mathcal{H}_1$ to obtain a HIBE which is secure in model $\boldsymbol{\mathcal{M}}_1$ and which supports key delegation over any number of levels. The required modifications are as follows.

- The public parameters are $(P, P_1, P_2, P_3, Q_1, \ldots, Q_n)$.
- $V(y) = y^n Q_n + \cdots + yQ_1 + P_3$ and $V_i = V(\mathsf{v}_i)$ as in the case of $\mathcal{H}_1$.

With the above two changes, the rest of key generation, encryption and decryption are as in $\mathcal{H}_1$.

More specifically, let us look at key generation. The private key d_{v} corresponding to $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$ is defined to be

$$(xP_2 + r_1 V_1 + \ldots + r_\tau V_\tau, r_1 P, \ldots, r_\tau P) = (d_0, d_1, \ldots, d_\tau)$$

where $r_1, \ldots, r_\tau$ are random elements of Z_p.

Since the maximum number of levels is not fixed in the set-up phase, the HIBE supports unbounded key delegation. This HIBE can be proved to be secure in model $\boldsymbol{\mathcal{M}}_1$. However, it is not secure in the sID-model, the reason being the following. Note that the first component d_0 of the secret key does not depend upon the ordering of the components of v. Hence, for any permutation of the components of v, the first component remains the same and thus, one can obtain a valid private key for any permutation of the components of v. In the sID model, the adversary can commit to an ID v^* and then ask the key extraction oracle for a private key of v' which is a permutation of v^*. Using the obtained private key of v', the adversary can easily obtain a private key for v^* and hence decrypt the challenge ciphertext. This shows insecurity in the sID model. Since, sID model is an accepted notion of security, insecurity in this model makes the unbouded depth HIBE less interesting and hence we will not consider this HIBE any further in this paper.

5.2 HIBE $\mathcal{H}_2$

The description of $\mathcal{H}_2$ is similar to that of $\mathcal{H}_1$. The differences are in the specification of the public parameters and the definition of the V_i's.

1. Let $(n_1, \ldots, n_h)$ be a tuple of positive integers.
2. The new public parameters are $(P, P_1, P_2, \overrightarrow{P}_3, \overrightarrow{Q}_1, \ldots, \overrightarrow{Q}_h)$ where $\overrightarrow{P}_3 = (P_{3,1}, \ldots, P_{3,h})$ and $\overrightarrow{Q}_i = (Q_{i,1}, \ldots, Q_{i,n_i})$. The master secret is αP_2.
3. Define
$$V_i(y) = y^{n_i} Q_{i,n_i} + y^{n_i-1} Q_{i,n_i-1} + \ldots + y Q_{i,1} + P_{3,i}.$$
As before V_i is used to denote $V_i(\mathsf{v}_i)$.

With these differences, the rest of set-up, key generation, encryption and decryption algorithms remain the same.

Note: The HIBE $\mathcal{H}_1$ has the parameters h and n and we will write (h, n)-$\mathcal{H}_1$ to denote this explicit parametrization. The HIBE $\mathcal{H}_2$ is parametrized by the tuple $(n_1, \ldots, n_h)$ and we will write $(h, n_1, \ldots, n_h)$-$\mathcal{H}_2$ to denote this parametrization.

6 Security Reduction

In this section, we show security reductions for the HIBE protocols.

Recall that the security models $\mathcal{M}_1$ and $\mathcal{M}_2$ are parametrized as (h, n)-$\mathcal{M}_1$ and $(h, n_1, \ldots, n_h)$-$\mathcal{M}_2$. The advantage of an adversary in the security game is denoted by Adv. A subscript to this will denote the model and a superscript will denote the HIBE for which the result is being stated. For example, $\mathsf{Adv}^{(h,n)\text{-}\mathcal{H}_1}_{(h,n)\text{-}\mathcal{M}_1}(t, q)$ denotes the maximum advantage of any adversary running in time t and making q queries to $\mathcal{O}_k$ in winning the security game defined by (h, n)-$\mathcal{M}_1$ for the HIBE (h, n)-$\mathcal{H}_1$. We will assume that one scalar multiplication in G_1 can be done in time $O(\sigma)$.

6.1 Security Reduction for $\mathcal{H}_1$

Theorem 1. *Let h, n, q be positive integers and n' be another positive integer with $n' \leq n$. Then*
$$\mathsf{Adv}^{(h,n)\text{-}\mathcal{H}_1}_{(h,n')\text{-}\mathcal{M}_1}(t, q) \leq \mathsf{Adv}^{\mathsf{DBDH}}(t + O(\sigma n q)).$$

Proof: The security reduction is to show that if there is an adversary which can break $\mathcal{H}_1$ then one obtains an algorithm to solve DBDH. The heart of such an algorithm is a simulator which is constructed as follows. Given an instance of DBDH as input, the simulator plays the security game (h, n')-$\mathcal{M}_1$ with an adversary for (h, n)-$\mathcal{H}_1$. The adversary executes the commitment stage; then the simulator sets up the HIBE based on the adversary's commitment as well as the DBDH instance.

The simulator gives the public parameters to the adversary and continues the game by answering all queries made by the adversary. In the process, it randomly chooses a bit γ and encrypts M_γ using the DBDH instance provided as input. Finally, the adversary outputs γ'. Based on the value of γ and γ', the simulator decides whether the instance it received is real or random. Intuitively, if the adversary has an advantage in breaking the HIBE protocol, the simulator also has an advantage in distinguishing between real and random instances. This leads to an upper bound on the advantage of the adversary in terms of the advantage of the simulator in solving DBDH.

We want to prove (h,n)-$\mathcal{H}_1$ secure in model (h,n')-$\boldsymbol{\mathcal{M}}_1$, where $1 \le n' \le n$. This means that the public parameters of the HIBE depend on n, while the adversary commits to a set $\mathcal{I}^*$ of size n' in the commit phase.

DBDH Instance: The simulator receives an instance $(P, P_1 = aP, P_2 = bP, Q = cP, Z)$ of DBDH.

The simulator now starts the security game for model $\boldsymbol{\mathcal{M}}_1$. This consists of several stages which we describe below. We will consider security against chosen plaintext attacks and hence the adversary will only have access to the key extraction oracle $\mathcal{O}_k$.

Adversary's Commitment: The adversary commits to a set $\mathcal{I}^*$ of size n'. The elements of $\mathcal{I}^*$ are from $\mathbb{Z}_p$. We write $\mathcal{I}^* = \{\mathsf{v}_1^*, \ldots, \mathsf{v}_{n'}^*\}$.

Set-Up: Define a polynomial $F(x)$ in $\mathbb{Z}_p[x]$ as follows.

$$\begin{aligned} F(x) &= (x - \mathsf{v}_1^*) \cdots (x - \mathsf{v}_{n'}^*) && (1)\\ &= x^{n'} + a_{n'-1}x^{n'-1} + \cdots + a_1 x + a_0 && (2) \end{aligned}$$

where the coefficients a_i's are in $\mathbb{Z}_p$ and are obtained from the values $\{\mathsf{v}_1^*, \ldots, \mathsf{v}_{n'}^*\}$. Since $F(x)$ is a polynomial of degree n' over $\mathbb{Z}_p$ and $\mathsf{v}_1^*, \ldots, \mathsf{v}_{n'}^*$ are its n distinct roots, we have $F(y) \neq 0$ for any $y \in \mathbb{Z}_p \setminus \{\mathsf{v}_1^*, \ldots, \mathsf{v}_{n'}^*\}$. The coefficients of $F(x)$ depend on the adversary's input and one cannot assume any distribution on these values. Define $a_{n'} = 1$ and $a_n = a_{n-1} = \cdots = a_{n'+1} = 0$.

For $1 \le i \le h$, define another set of polynomials $J_i(x)$ each of degree n in the following manner. Randomly choose $b_{0,1}, \ldots, b_{0,h}, b_1, \ldots, b_n$ from $\mathbb{Z}_p$. Define

$$J_i(x) = b_n x^n + b_{n-1}x^{n-1} + \cdots + b_1 x + b_{0,i} \qquad (3)$$

The public parameters $P_{3,i}$s and Q_js are defined in the following manner.

- For $1 \le i \le h$, define $P_{3,i} = a_0 P_2 + b_{0,i} P$.
- For $1 \le j \le n$, define $Q_j = a_j P_2 + b_j P$.

Since $b_{0,1}, \ldots, b_{0,h}, b_1, \ldots, b_n$ are chosen randomly from $\mathbb{Z}_p$, the $P_{3,i}$s and the Q_js are random elements of G_1. The public parameters are given to the adversary. The master secret is aP_2, which is not known to the simulator.

Now comes the most crucial part of the proof. For $y \in \mathbb{Z}_p$,

$$\left.\begin{aligned} V_i(y) &= P_{3,i} + yQ_1 + y^2 Q_2 + \cdots + y^n Q_n \\ &= a_0 P_2 + b_{0,i} P + y(a_1 P_2 + b_1 P) + y^2(a_2 P_2 + b_2 P) + \cdots + y^n(a_n P_2 + b_n P) \\ &= (a_0 + a_1 y + a_2 y^2 + \cdots + a_n y^n) P_2 + (b_{0,i} + b_1 y + b_2 y^2 + \cdots + b_n y^n) P \\ &= F(y) P_2 + J_i(y) P. \end{aligned}\right\} \qquad (4)$$

This decomposes $V_i(y)$ into two parts – one depends on P_2 and the other depends on P. The part which depends on P_2 vanishes if and only if y is equal to some element of $\mathcal{I}^*$. The ability of the simulator to properly answer key extraction queries and generate a proper challenge ciphertext depends crucially on this fact.

Phase 1: In this stage, the adversary can make queries to $\mathcal{O}_k$, all of which have to be answered by the simulator. Suppose the adversary queries $\mathcal{O}_k$ on an identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$, with $1 \le \tau \le h$. By the constraint of model $\boldsymbol{\mathcal{M}}_1$ all the v_i's cannot be in $\mathcal{I}^*$. Suppose $\imath$ is such that $\mathsf{v}_\imath$ is not in $\mathcal{I}^*$. Then $F(\mathsf{v}_\imath) \not\equiv 0 \bmod p$.

As in the protocol, define V_i to be $V_i(\mathsf{v}_i)$. Choose $r_1, \ldots, r_{\imath-1}, r'_\imath, r_{\imath+1}, \ldots, r_\tau$ randomly from $\mathbb{Z}_p$. Define $W = \sum_{i=1, i\neq\imath}^{\tau} r_i V_i$. The first component d_0 of the secret key for $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$ is computed in the following manner.

$$d_0 = -\frac{J_\imath(v_\imath)}{F(v_\imath)} P_1 + r'_\imath (F(v_\imath) P_2 + J_\imath(v_\imath) P) + W.$$

The following computation shows that d_0 is properly formed.

$$\begin{aligned} d_0 &= \pm a P_2 - \frac{J_\imath(v_\imath)}{F(v_\imath)} P_1 + r'_\imath (F(v_\imath) P_2 + J_\imath(v_\imath) P) + W \\ &= a P_2 + \left(r'_\imath - \frac{a}{F(v_\imath)} \right) (F(v_\imath) P_2 + J_\imath(v_\imath) P) + W \\ &= a P_2 + \sum_{i=1}^{\tau} r_i V_i \end{aligned}$$

where $r_\imath = r'_\imath - a/F(v_\imath)$. Since $r'_\imath$ is random, so is $r_\imath$. The quantities $d_1, \ldots, d_\tau$ are computed in the following manner.

$$\begin{aligned} d_i &= r_i P & & 1 \le i \le \tau,\ i \neq \imath; \\ &= r'_\imath P - \tfrac{1}{F(v_\imath)} P_1 = r_\imath P & & \text{for } i = \imath. \end{aligned}$$

This technique is based on the algebraic techniques introduced by Boneh and Boyen [3]. The generalization is in the definition of $F()$ and $J_i()$s. Here we take these to be polynomials, which allows us to tackle the case of adversary committing to more than one identity.

Challenge Generation: The adversary submits messages M_0, M_1 and an identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$ with $1 \le \tau \le h$. By the rules of model $\boldsymbol{\mathcal{M}}_1$, each $\mathsf{v}_i \in \mathcal{I}^*$ and so $F(\mathsf{v}_i) \equiv 0 \bmod p$ for $1 \le i \le \tau$. Consequently,

$$V_i = V_i(\mathsf{v}_i) = F(\mathsf{v}_i) P_2 + J_i(\mathsf{v}_i) P = J_i(\mathsf{v}_i) P$$

and hence

$$c V_i = c J_i(\mathsf{v}_i) P = J_i(\mathsf{v}_i)(cP) = J_i(\mathsf{v}_i) Q$$

where $Q = cP$ was supplied as part of the DBDH instance. Note that it is possible to compute $W_i = cV_i$ even without knowing c. The simulator now randomly chooses a bit γ and returns

$$(M_\gamma \times Z, Q, W_1, \ldots, W_\tau)$$

to the adversary. If Z is real, then this is a proper encryption of M_γ under the identity v.

Phase 2: The key extraction queries in this stage are handled as in Phase 1.

Guess: The adversary outputs a guess γ'. The simulator outputs 1 if $\gamma = \gamma'$, else it outputs 0.

If $Z = e(P, P)^{abc}$, then the simulator provides a perfect simulation of the (h, n')-$\boldsymbol{\mathcal{M}}_1$ game. On the other hand, if Z is random, the adversary receives no information about the message M_γ from the challenge ciphertext.

The above shows that an adversary's ability to attack (h, n)-$\mathcal{H}_1$ HIBE in model (h, n')-$\boldsymbol{\mathcal{M}}_1$ can be converted into an algorithm for solving DBDH. The bound on the advantage follows from this fact. □

Theorem 1 shows that an (h, n)-$\mathcal{H}_1$ HIBE is CPA-secure in model (h, n')-$\boldsymbol{\mathcal{M}}_1$ for $n' \leq n$. The next result shows that (h, n)-$\mathcal{H}_1$ is also secure in the h-sID model.

Theorem 2. *Let h, n, q be positive integers. Then*

$$\mathsf{Adv}_{h\text{-}sID}^{(h,n)\text{-}\mathcal{H}_1}(t, q) \leq \frac{q}{p} + \mathsf{Adv}^{\mathsf{DBDH}}(t + O(\sigma n q)).$$

Proof: The proof is similar to the proof of Theorem 1. In the h-sID model, the adversary commits to an identity $(\mathsf{v}_1^*, \ldots, \mathsf{v}_\tau^*)$ where $1 \leq \tau \leq h$ and $\mathsf{v}_i \in \mathbb{Z}_p$. Randomly choose $\mathsf{v}_{\tau+1}^*, \ldots, \mathsf{v}_h^*$ from $\mathbb{Z}_p$. Randomly choose $b_1, \ldots, b_n, b_{0,1}, \ldots, b_{0,h}$ from $\mathbb{Z}_p$. For $1 \leq i \leq h$, define

$$\begin{aligned} F_i(x) &= x - \mathsf{v}_i^*; \\ J_i(x) &= b_n x^n + b_{n-1} x^{n-1} + \cdots + b_1 x + b_{0,i}. \end{aligned}$$

The protocol is set-up in the following manner. For $2 \leq j \leq n$, define $Q_j = b_j P$, $Q_1 = P_2 + b_1 P$ and for $1 \leq i \leq h$, define $P_{3,i} = -v_i^* P_2 + b_{0,i} P$. This defines all the public parameters.

For $1 \leq i \leq h$, we have

$$\begin{aligned} V_i(y) &= P_{3,i} + y Q_1 + y^2 Q_2 + \cdots + y^n Q_n \\ &= (-v_i^* P_2 + b_{0,i} P) + y(P_2 + b_1 P) + y^2 b_2 P + \cdots + y^n b_n P \\ &= (y - v_i^*) P_2 + (b_{0,i} + b_1 y + \cdots + b_n y^n) P \\ &= F_i(y) P_2 + J_i(y) P \end{aligned}$$

The rest of the simulation is similar to the proof of Theorem 1 with one difference. If the adversary ever submits a key extraction query of the form $(\mathsf{v}_1, \ldots, \mathsf{v}_j)$, with $j > \tau$ and $\mathsf{v}_i = \mathsf{v}_i^*$ for $1 \leq i \leq j$, then the simulator aborts and outputs a random bit. Note that since the length of the identity is longer than the committed identity, the adversary is allowed to make such queries. The probability that $\mathsf{v}_i = \mathsf{v}_i^*$ for $\tau < i \leq j$ is $1/p^{j-\tau} \leq 1/p$. Since this can be repeated for each of the q key extraction queries, we have the additive degradation by the factor q/p. □

6.2 Security Reduction for $\boldsymbol{\mathcal{H}_2}$

Theorem 3. *Let $h, n_1, \ldots, n_h, q$ be positive integers and $n'_1, \ldots, n'_h$ be another set of positive integers with $n'_i \leq n_i$ for $1 \leq i \leq h$. Then*

$$\mathsf{Adv}_{(h,n'_1,\ldots,n'_h)\text{-}\boldsymbol{\mathcal{M}}_2}^{(h,n_1,\ldots,n_h)\text{-}\mathcal{H}_2}(t, q) \leq \mathsf{Adv}^{\mathsf{DBDH}}(t + O(\sigma n q))$$

where $n = \sum_{i=1}^{h} n_i$.

Proof: The proof is similar to the proof of Theorem 1. The difference is in the definition of $F(x)$ and $J_i(x)$. In this case, for $1 \leq i \leq h$, we require $F_i(x)$. After the appropriate definition of $F_i(x)$ and $J_i(x)$, we show that $V_i(y)$ can be written as $V_i(y) = F_i(y)P_2 + J_i(y)P$. As mentioned in the proof of Theorem 1, the simulator's ability for answering key extraction queries and challenge generation depends upon this decomposition. The actual procedure for key extraction and challenge generation is very similar to that in the proof of Theorem 1 and these details are not provided. Thus, we provide the details of only two stages of the game – adversary's commitment and set-up.

The simulator is given an instance $(P, P_1 = aP, P_2 = bP, Q = cP, Z)$ of DBDH.

Adversary's Commitment: Following model $(h, n'_1, \ldots, n'_h)$-$\boldsymbol{\mathcal{M}}_2$, the adversary commits to sets $\mathcal{I}^*_1, \ldots, \mathcal{I}^*_\tau$, where $|\mathcal{I}^*_i| = n'_i$ and $1 \leq \tau \leq h$.

Set-Up: The simulator defines polynomials $F_1(x), \ldots, F_h(x)$, and $J_1(x), \ldots,$ $J_h(x)$ in the following manner. For $1 \leq i \leq \tau$, define

$$\begin{aligned} F_i(x) &= \prod_{\mathsf{v} \in \mathcal{I}_i} (x - \mathsf{v}) \\ &= x^{n'_i} + a_{i,n'_i-1}x^{n'_i-1} + \cdots + a_{i,1}x + a_{i,0}; \end{aligned}$$

For $\tau+1 \leq i \leq h$ choose a random non-zero element $a_{i,0}$ from $\mathbb{Z}_p$ and define $F_i(x) = a_{i,0}$. Note that $F_i(x)$ is a non-zero constant polynomial. For $1 \leq i \leq \tau$, define $a_{i,n'_i} = 1$ and $a_{i,n_i} = \cdots = a_{i,n'_i+1} = 0$; for $\tau + 1 \leq i \leq h$, set $n'_i = 0$ and $a_{i,1} = \cdots = a_{i,n_i} = 0$.

For $1 \leq i \leq h$ and $1 \leq j \leq n_i$ choose random elements $b_{i,j}$ from $\mathbb{Z}_p$. Define

$$J_i(x) = b_{i,n_i}x^{n_i} + b_{i,n_i-1}x^{n_i-1} + \cdots + b_{i,1}x + b_{i,0}.$$

Note that $F_i(x)$ is of degree n'_i while $J_i(x)$ is of degree n_i.

The public parameters are defined as follows.

- For $1 \leq i \leq h$, define $P_{3,i} = a_{i,0}P_2 + b_{i,0}P$.
- For $1 \leq i \leq h$ and $1 \leq j \leq n_i$ define $Q_{i,j} = a_{i,j}P_2 + b_{i,j}P$.

Since the $b_{i,j}$s are chosen randomly, the distribution of the public parameters is random. We now show the decomposition of $V_i(y)$.

$$\left.\begin{aligned} V_i(y) &= P_{3,i} + yQ_{i,1} + y^2Q_{i,2} + \cdots + y^{n_i}Q_{i,n_i} \\ &= (a_{i,0}P_2 + b_{i,0}P) + y(a_{i,1}P_2 + b_{i,1}P) + y^2(a_{i,2}P_2 + b_{i,2}P) + \cdots + y^{n_i}(a_{i,n_i}P_2 + b_{i,n_i}P) \\ &= (a_{i,0} + a_{i,1}y + a_{i,2}y^2 + \cdots + a_{i,n_i}y^{n_i})P_2 + (b_{i,0} + b_{i,1}y + b_{i,2}y^2 + \cdots + b_{i,n_i}y^{n_i})P \\ &= F_i(y)P_2 + J_i(y)P. \end{aligned}\right\} \quad (5)$$

The rest of the simulation is very similar to that in the proof of Theorem 1. Also, the bound on the advantage follows as in the above mentioned proof. □

The proof shows that $(h, n_1, \ldots, n_h)$-$\mathcal{H}_2$ is secure in model $(h, n'_1, \ldots, n'_h)$-$\boldsymbol{\mathcal{M}}_2$ with $n'_i \leq n_i$. Recall that $(h, 1, \ldots, 1)$-$\boldsymbol{\mathcal{M}}_2$ is same as the h-sID model and hence $(h, n_1, \ldots, n_h)$-$\mathcal{H}_2$ is secure in the h-sID model.

7 Multi-Receiver IBE

A multi-receiver IBE (MR-IBE) is an extension of the IBE, which allows a sender to encrypt a message in such a way that it can be decrypted by any one of a particular set of identities. In other words, there is one encryptor but more than one valid receivers. In IBE, the number of valid receivers is one. One trivial way to realize an MR-IBE from an IBE is to encrypt the same message several times. A non-trivial construction attempts to reduce the cost of encryption.

This notion was introduced in [1] and a non-trivial construction based on the Boneh-Franklin IBE (BF-IBE) was provided. The construction was proved to be secure in the *selective*-ID model under the *random oracle* assumption. Note that the BF-IBE is secure in the full model under the random oracle assumption.

We show that $\mathcal{H}_1$ restricted to IBE can be modified to obtain an MR-IBE. The situation for $\mathcal{H}_1$ is almost identical. The required modifications to the protocol are as follows.

1. The encryption is converted into a hybrid scheme. Instead of multiplying the message with the "mask" $Z = e(P_1, P_2)^t$, the value Z is provided as input to a pseudorandom generator and the message (considered to be a bit string) is XORed with the resulting keystream.
2. The private key corresponding to an identity v is $d_{\mathsf{v}} = (xP_2 + rV_{\mathsf{v}}, rP)$, where $V_{\mathsf{v}} = P_{3,1} + V(\mathsf{v})$ as defined in in Section 5.1.
3. Suppose the intended set of receivers is $\{\mathsf{v}_1, \ldots, \mathsf{v}_\tau\}$. Then the ciphertext consists of the encryption of the message as mentioned above plus a header of the form $(tP, tV_1, \ldots, tV_\tau)$, where V_i is as defined in the construction of $\mathcal{H}_1$ in Section 5.1 and t is a random element of Z_p.
4. The receiver possessing the secret key d_{v_i} $(1 \leq i \leq \tau)$ can compute $e(P_1, P_2)^t$ in the standard manner and hence obtain the input to the pseudorandom generator. Thus it can decrypt the message.

The MR-IBE described above can be proved to be secure in the selective-ID model *without* the random oracle assumption. The security model for MR-IBE is the following. In the commitment stage, the adversary commits to a set of identities; does not ask for the private key of these identities in the key extraction queries and finally asks for the encryption under this set of identities. Note that this is very similar to the model $\boldsymbol{\mathcal{M}}_1$ restricted to IBE. The only difference is that during the generation of the challenge ciphertext, in $\boldsymbol{\mathcal{M}}_1$, the adversary supplies only one identity out of the set of identities it had previously committed to, whereas in the model for MR-IBE, the adversary asks for the encryption under the whole set of these identities.

This difference is easily tackled in our proof in Section 6.1 which shows that $\mathcal{H}_1$ is secure in model $\boldsymbol{\mathcal{M}}_1$. Recall that the construction of the polynomial $F(x)$ is such that $F(\mathsf{v}) = 0$ for all $\mathsf{v} \in \mathcal{I}^*$, where $\mathcal{I}^*$ is the set of committed identities. In the challenge stage of the security proof for $\mathcal{H}_1$ as an IBE, we use this fact for only one identity (the identity given by the adversary). In the proof for MR-IBE, we will need to generate cV_i for all $\mathsf{v} \in \mathcal{I}^*$. Since $F(\mathsf{v}) = 0$ for any such v, this can be done in the standard fashion.

The above argument does not provide any security degradation. Hence, we obtain an MR-IBE which can be proved to be secure in the selective-ID model *without* the random oracle assumption.

8 Conclusion

In this paper, we have generalized the notion of *selective*-ID secure HIBE. Two new security models $\boldsymbol{\mathcal{M}}_1$ and $\boldsymbol{\mathcal{M}}_2$ have been introduced. In the security game, both these models allow an adversary to commit to a set of identities (as opposed to a single identity in the sID model) before the set-up. During the challenge stage, the adversary can choose any one of the previously committed identities as a challenge identity. We provide two HIBE constructions $\mathcal{H}_1$ and $\mathcal{H}_2$ which are secure in the models $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively. The public parameter size is smaller in case of $\mathcal{H}_1$. Further, we also show that $\mathcal{H}_1$ and $\mathcal{H}_2$ can be modified to obtain an MR-IBE protocol which is secure in the sID model *without* random oracles. The only previous construction of MR-IBE is secure in the sID model under the random oracle assumption.

9 Acknowledgment

The authors express their sincere gratitude to the anonymous reviewers of PKC 2006.

References

1. Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. Efficient multi-receiver identity-based encryption and its application to broadcast encryption. In Serge Vaudenay, editor, *Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 380–397. Springer, 2005.
2. Paulo S. L. M. Barreto, Hae Yong Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 354–368. Springer, 2002.
3. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In Cachin and Camenisch [9], pages 223–238.
4. Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459. Springer, 2004.
5. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Cramer [13], pages 440–456.
6. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.
7. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.
8. Xavier Boyen, Qixiang Mei, and Brent Waters. Direct chosen ciphertext security from identity-based techniques. In Vijay Atluri, Catherine Meadows, and Ari Juels, editors, *ACM Conference on Computer and Communications Security*, pages 320–329. ACM, 2005.
9. Christian Cachin and Jan Camenisch, editors. *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*. Springer, 2004.
10. Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 255–271. Springer, 2003.
11. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Cachin and Camenisch [9], pages 207–222.
12. Sanjit Chatterjee and Palash Sarkar. Trading time for space: Towards an efficient IBE scheme with short(er) public parameters in the standard model. In *ICISC*, 2005.
13. Ronald Cramer, editor. *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*. Springer, 2005.
14. Steven D. Galbraith, Keith Harrison, and David Soldera. Implementing the Tate pairing. In Claus Fieker and David R. Kohel, editors, *ANTS*, volume 2369 of *Lecture Notes in Computer Science*, pages 324–337. Springer, 2002.
15. Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography. In Yuliang Zheng, editor, *ASIACRYPT*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer, 2002.
16. Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 466–481. Springer, 2002.
17. Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.
18. Brent Waters. Efficient identity-based encryption without random oracles. In Cramer [13], pages 114–127.