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. 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. One of the HIBE constructions supports an unbounded number of levels, i.e., the maximum number of levels does not need to be specified during the set-up. Further, we show that this HIBE 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 [16]. 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) [14, 15]. 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 [9]. This paper also introduced an appropriate security model for IBE. The proof of security in [9] 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 a HIBE was given in [11]. However, the HIBE in [11] can only be proved to be secure in a weaker model (the selective-ID model) as opposed to the full model considered in [9]. Later Boneh and Boyen [4] presented a more efficient construction of HIBE which is also secure in the selective-ID (sID) model without the random oracle assumption.

The full security model in [9] 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 prefix, 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 $\boldsymbol{\mathcal{M}}_1$ and $\boldsymbol{\mathcal{M}}_2$.

In $\boldsymbol{\mathcal{M}}_1$, 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, $\boldsymbol{\mathcal{M}}_2$, 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) [4]. 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 the case of $\mathcal{H}_2$, the number of required scalar multiplications increases linearly with the size of the committed set of identities.

One interesting feature about $\mathcal{H}_1$ is that it can support unbounded number of levels. In other words, the set-up for $\mathcal{H}_1$ does not specify the maximum number of levels of the HIBE. This is an added advantage and to the best of our knowledge is not present in any of the previous HIBE constructions.

The situation for $\mathcal{H}_1$ is also interesting in another aspect. If we consider only IBE, then the number of scalar multiplications increases with the size of the committed set of identities. On the other hand, in the case of BB-HIBE, the number of scalar multiplications increases linearly with the depth of the HIBE. Since $\mathcal{H}_1$ can support HIBE of unbounded depth, this feature is not present in $\mathcal{H}_1$.

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 $\mathcal{H}_1$ 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 Security Model for HIBE

2.1 HIBE Protocol

Following [15, 14] a hierarchical identity based encryption (HIBE) scheme is specified by four algorithms: Setup, Key Generation, Encryption and Decryption.

Setup: It takes 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. The system parameters may also specify a positive integer h, which denotes the maximum number of levels that are allowed in the HIBE. *If h is not specified, then the HIBE can support an unbounded number of levels.* 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 $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$ is done 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: It takes as input the identity v and a message from the message space and produces a ciphertext in the cipher space.

Decryption: It takes as input the ciphertext and the private key of the corresponding identity v and returns the message or bad if the ciphertext is not valid.

2.2 Security Model

The security model for HIBE is defined as an interactive game between an adversary and a simulator. 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 as we will define later.

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

There is a bit of 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 has to provide a proper decryption. 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 $\mathcal{S}_1$. 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 chooses an identity $\mathsf{v}^* \in \mathcal{S}_2$ with the restriction that it has not queried $\mathcal{O}_k$ for the private key of v^* or any of its prefixes and two messages M_0, M_1 and provides these to the simulator. 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 (obvious) restriction that it cannot ask $\mathcal{O}_d$ for the decryption of C^* under v^* nor $\mathcal{O}_k$ for the private key of any prefix of v^*.

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:

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

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 secure if $\mathsf{Adv}^{\mathsf{HIBE}}(t, q_{\mathsf{ID}}, q_{\mathsf{C}})$ is negligible. 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. There are several generic as well as non-generic methods for converting a CPA-secure HIBE into a CCA-secure HIBE. Hence, in this paper, we will only consider construction of CPA-secure HIBE.

2.3 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 other restriction that the adversary has not asked for the private key for any prefix of the challenge identity as well as the restriction in Phase 2 still applies.

2.4 Selective-ID Model

Let $\mathcal{S}_1 = \mathcal{S}_2$ be a singleton set. This means that the adversary commits to one particular identity; never asks for its private key; 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.

2.5 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. We define $\mathcal{S}_1 = \mathcal{S}_2$ to be the set of all tuples $(\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$, $(\tau \geq 1)$, such that each $\mathsf{v}_i \in \mathcal{I}^*$. First consider the case of IBE, i.e., where 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 2.4. In other words, for IBE protocol, $\boldsymbol{\mathcal{M}}_1$ is a strict generalization of sID model.

Let us now see what this means. In the commit phase, the adversary commits to the set of identities $\mathcal{I}^*$; never asks for the private key of any of these identities; and during the challenge phase presents one of these identities to the simulator. This is the generalization of the sID model, where instead of a single identity, the adversary may choose one from a set of identities.

In the case of HIBE, the situation is different. 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. We later discuss the interpretation of this model as well as the other ones.

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

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

This model is a strict generalization of the sID model for HIBE. This can be seen by setting $n_1 = \cdots = n_\tau = 1$, i.e., $\mathcal{I}_1^*, \ldots, \mathcal{I}_\tau^*$ to be singleton sets.

3 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 indistinguishability is derived 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 $\mathbf{e}$ in the HIBE structure, the adversary's behaviour 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 $\mathbf{e}$ or one of its ancestors, then it will not be able to succeed in an attack where the target identity is that of $\mathbf{e}$. 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 $\mathbf{e}$ 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 $\mathbf{e}$. Entity $\mathbf{e}$ 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, $\mathbf{e}$ 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 targetting 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 $\mathsf{sysadmin}$. This is a serious limitation of the sID model.

This brings us to the generalization of the sID model that we have introduced. First consider the model $\boldsymbol{\mathcal{M}}_1$ 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 a strictly better security than the sID model.

Now consider the model $\boldsymbol{\mathcal{M}}_1$ 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 $\boldsymbol{\mathcal{M}}_1$ provides an acceptable security notion for HIBE. Intuitively, it provides better security than the sID model for HIBE, though we cannot fix the parameters of $\boldsymbol{\mathcal{M}}_1$ so that it collapses to the sID model for HIBE.

The model $\boldsymbol{\mathcal{M}}_2$ is a clear generalization of the usual sID model for HIBE. The adversary fixes the sensitive keywords for each level of the HIBE upto 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 difference between models $\boldsymbol{\mathcal{M}}_1$ and $\boldsymbol{\mathcal{M}}_2$ is that from a technical point of view, in $\boldsymbol{\mathcal{M}}_2$, for each level of the HIBE, the adversary is allowed to indepedently choose the set of possible values which the corresponding component of the target identity may take. In $\boldsymbol{\mathcal{M}}_1$, the set of possible values for all components are the same. It is due to this difference, that we cannot collapse $\boldsymbol{\mathcal{M}}_1$ to the sID model. On the other hand, in practical applications, the sensitive keywords for all levels are likely to be the same. In such a situation, $\boldsymbol{\mathcal{M}}_1$ provides a more cleaner notion of security. Of course, this is still much less comprehensive than the full security model.

4 Constructions

We present two HIBE protocols $\mathcal{H}_1$ and $\mathcal{H}_2$. The HIBE $\mathcal{H}_1$ can be proved to be secure in model $\boldsymbol{\mathcal{M}}_1$, whereas the HIBE $\mathcal{H}_2$ can be proved to be secure in model $\boldsymbol{\mathcal{M}}_2$.

4.1 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 $e : G_1 \times G_1 \rightarrow G_2$ 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 $a, b \in Z_p$.
- 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. Modified Weil pairing [8] and Tate pairing [2, 13] are examples of cryptographic bilinear maps.

4.2 HIBE $\mathcal{H}_1$

Set-Up: The identity space consists of all tuples $(\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$, where each $\mathsf{v}_i \in Z_p$. Note that we do not fix a upper bound on τ. 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 $x \in Z_p$ and set $P_1 = xP$. Randomly choose P_2, P_3, $Q_1, \ldots, Q_n$ from G_1 where n is a parameter of the model. The public parameters are $(P, P_1, P_2, P_3, Q_1, \ldots, Q_n)$ and the master secret key is xP_2. Note that, the public parameter size does not depend on the levels of the HIBE. In other words, potentially $\mathcal{H}_1$ can support unbounded number of levels. Since, P_1, P_2 are not directly required in Encryption or Decryption, we may replace them in the public parameters by $e(P_1, P_2)$. This will save the pairing computation during the encryption.

Key Generation: Let $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$ be an identity. For any $y \in Z_p$ define

$$V(y) = y^n Q_n + \cdots + y Q_1.$$

Let $V_i = P_3 + V(\mathsf{v}_i)$. The private key d_v corresponding to v 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. It is standard [4] to verify 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

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

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

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

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

4.3 HIBE $\mathcal{H}_2$

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

1. Define the maximum depth of the HIBE to be h. Additionally, a tuple $(n_1, \ldots, n_h)$ of positive integers is required.
2. Replace P_3 in $\mathcal{H}_1$, by the tuple $(P_{3,1}, \ldots, P_{3,h})$ where each $P_{3,i}$ is an element of G_1. Also the points Q_i's $(1 \le i \le n)$ are replaced by the points $Q_{i,j}$'s, where $1 \le i \le h$ and $1 \le j \le n_i$.
3. Define $V(i, y) = y^{n_i} Q_{i,n_i} + \ldots + y Q_{i,1}$. Given an identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$, define $V_i = P_{3,i} + V(i, \mathsf{v}_i)$.

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

5 Security Reduction

In this section, we show that the breaking of $\mathcal{H}_1$ amounts to solving the DBDH problem and similarly for $\mathcal{H}_2$.

5.1 Hardness Assumption

Assume the bilinear map notation from Section 4.1. The DBDH problem in $G_1, G_2, e()$ [9] 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 is real or Z is random. The advantage of a probabilistic algorithm $\mathcal{B}$, which takes as input a tuple $\langle P, aP, bP, cP, Z\rangle$ and outputs a bit, in solving the DBDH problem is defined as

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

where the probability is calculated over the random choice of $a, b, c \in Z_p$ as well as the random bits used by $\mathcal{B}$. The quantity $\mathsf{Adv}^{\mathsf{DBDH}}(t)$ denotes the maximum of $\mathsf{Adv}^{\mathsf{DBDH}}_{\mathcal{B}}$ where the maximum is taken over all adversaries running in time at most t.

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

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. On given an instance of DBDH as input, the simulator plays the security game with an adversary for $\mathcal{H}_1$. 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 guesses the 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 a good 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. The details of the reduction are given below.

DBDH Instance: The simulator receives an instance $(P, P_1 = aP, P_2 = bP, Q = cP, Z) \in G_1^4 \times G_2$ of DBDH. It has to decide whether $Z = e(P, P)^{abc}$ (i.e., Z is real) or whether Z is random. Note that it does not know a, b, c.

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 $\mathcal{I}^*$. We will assume that the elements of $\mathcal{I}^*$ are elements of Z_p. Alternatively, if these are bit strings, then (as is standard) they will be hashed using a collision resistant hash function into elements of Z_p. We write $\mathcal{I}^* = \{\mathsf{v}_1^*, \ldots, \mathsf{v}_n^*\}$.

Set-Up: Define a polynomial in $Z_p[x]$ by

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

where the coefficients a_i's are in 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 Z_p and $\mathsf{v}_1^*, \ldots, \mathsf{v}_n^*$ are its n distinct roots, we have $F(\mathsf{v}) \neq 0$ for any $\mathsf{v} \in Z_p \setminus \{\mathsf{v}_1^*, \ldots, \mathsf{v}_n^*\}$.) Note that, these coefficients depend on the adversary's input and one cannot assume any distribution on these values. For notational convenience, we define $a_n = 1$. Randomly choose $b_0, \ldots, b_n$ from Z_p and define another polynomial

$$J(x) = b_n x^n + b_{n-1}x^{n-1} + \cdots + b_1 x + b_0 \qquad (3)$$

Define $P_3 = a_0 P_2 + b_0 P$ and for $1 \leq i \leq n$, define $Q_i = a_i P_2 + b_i P$. Note that, Q_is are random elements of G_1. Now note that for $y \in Z_p$,

$$\begin{aligned} V(y) &= P_3 + yQ_1 + y^2 Q_2 + \cdots + y^n Q_n \\ &= F(y)P_2 + J(y)P. \end{aligned}$$

The public parameters are $(P, P_1, P_2, P_3, Q_1, \ldots, Q_n)$ which has the same distribution as the public parameters in the protocol specification. These are given to the adversary. The master secret is aP_2, which is not known to the simulator.

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 $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$. 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 $F(\mathsf{v}_\imath) \neq 0$.

As in the protocol, define $V_i = P_3 + V(\mathsf{v}_i)$. Choose $r_1, \ldots, r_{\imath-1}, r'_\imath, r_{\imath+1}, \ldots, r_\tau$ randomly from 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.

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

The following computation shows that d_0 is a properly formed.

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

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.

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

This technique is based on the algebraic techniques introduced by Boneh and Boyen [4]. The generalization is in the definition of $F()$ and $J()$. Here we take these to be polynomials, which allows us to tackle the case of adversary committing to more than one identity. In case the polynomials are of degree one, then we get exactly the Boneh-Boyen HIBE [4].

Challenge Generation: The adversary submits messages M_0, M_1 and an identity $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\tau)$. By the rules of model $\boldsymbol{\mathcal{M}}_1$, each $\mathsf{v}_i \in \mathcal{I}^*$ and so $F(\mathsf{v}_i) = 0$ for $1 \leq i \leq \tau$. Consequently, $V_i = F(\mathsf{v}_i) P_2 + J(\mathsf{v}_i) P = J(\mathsf{v}_i) P$ and $c V_i = c J(\mathsf{v}_i) P = J(\mathsf{v}_i)(cP) = J(\mathsf{v}_i) Q = W_i$ (say), where $Q = cP$ was supplied as part of the DBDH instance. Note that it is possible to compute 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. 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 $\boldsymbol{\mathcal{M}}_1$ game. On the other hand, if Z is random, the adversary receives no information about the message M_γ from the challenge ciphertext. Formalizing this argument in the standard manner shows that $\mathsf{Adv}_{\mathcal{A}}^{\mathcal{H}_1}(t,q) \leq \mathsf{Adv}_{\mathcal{B}}^{\mathrm{DBDH}}(t + O(\sigma n q))$ where σ is the time for scalar multiplication in G_1 and q is the maximun number of queries allowed to the adversary.

5.3 Security Reduction for $\mathcal{H}_2$

The security reduction for $\mathcal{H}_2$ in model $\boldsymbol{\mathcal{M}}_2$ is similar to that of $\mathcal{H}_1$ in model $\boldsymbol{\mathcal{M}}_1$. We mention only the differences.

Adversary's Commitment: Following model $\boldsymbol{\mathcal{M}}_2$, the adversary commits to sets $\mathcal{I}_1^*, \ldots, \mathcal{I}_\tau^*$, where $|\mathcal{I}_i^*| = n_i$.

Set-Up: The simulator defines polynomials $F_1(x), \ldots, F_\tau(x)$, and $J_1(x), \ldots, J_\tau(x)$ where

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

where $b_{i,j}$'s are random elements of Z_p. For notational convenience, we define $a_{i,n_i} = 1$. For $1 \leq i \leq \tau$, define $P_{3,i} = a_{i,0}P_2 + b_{i,0}P$ and $Q_{i,j} = a_{i,j}P_2 + b_{i,j}P$, $1 \leq j \leq n_i$.

Key Extraction Query: Suppose the private key of $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_\nu)$ is required. According to model $\boldsymbol{\mathcal{M}}_2$, there is at least one i such that $\mathsf{v}_i \notin \mathcal{I}_i^*$. Then this i can be used to generate the private key in a manner similar to the key generation by the simulator for $\mathcal{H}_1$ in model $\boldsymbol{\mathcal{M}}_1$.

Challenge Generation: Suppose the challenge identity is $\mathsf{v}^* = (\mathsf{v}_1^*, \ldots, \mathsf{v}_\nu^*)$. Then by the constraint of $\boldsymbol{\mathcal{M}}_2$ for each i, $\mathsf{v}_i^* \in \mathcal{I}_i^*$ and consequently $F_i(\mathsf{v}_i^*) = 0$. This allows the generation of a proper ciphertext as in the simulation of $\mathcal{H}_1$ in model $\boldsymbol{\mathcal{M}}_1$.

Finally, we obtain the following result.

$$\mathsf{Adv}_{\mathcal{A}}^{\mathcal{H}_2}(t,q) \leq \mathsf{Adv}_{\mathcal{B}}^{\mathrm{DBDH}}(t + O(\sigma \sum_{i=1}^{h} n_i q)).$$

6 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 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 $V_{\mathsf{v}} = P_3 + V(\mathsf{v})$ as defined in in Section 4.2.
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 4.2 and t is a random element of Z_p.
4. The receiver possessing the secret key d_{v_i} $(1 \le i \le \tau)$ 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 $\mathcal{M}_1$ restricted to IBE. The only difference is that during the generation of the challenge ciphertext, in $\mathcal{M}_1$, 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 5.2 which shows that $\mathcal{H}_1$ is secure in model $\mathcal{M}_1$. 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.

7 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 $\boldsymbol{\mathcal{M}}_1$ and $\boldsymbol{\mathcal{M}}_2$ respectively. Interestingly, the HIBE $\mathcal{H}_1$ allows delegation of an unbounded number of levels, i.e., the maximum number of delegation levels is not fixed during the protocol set-up. Further, we also show that $\mathcal{H}_1$ 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.

8 Acknowledgement

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

References

1. J. Baek, R. Safavi-Naini and W. Susilo. Efficient Multi-Receiver Identity-Based Encryption and Its Application to Broadcast Encryption. PKC 2005, LNCS 3386, pp 380–397, 2005.
2. P. S. L. M. Barreto, H. Y. Kim, B. Lynn and M. Scott. Efficient Algorithms for Pairing-Based Cryptosystems. CRYPTO 2002, LNCS 2442, pp. 354–368, 2002.
3. M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In ACM Conference on Computer and Communications Security - CCS 1993, pp 62–73, 1993.
4. D. Boneh, X. Boyen. Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles, EUROCRYPT 2004, LNCS 3027, pp 223–238, 2004.
5. D. Boneh, X. Boyen. Secure Identity Based Encryption without Random Oracles. CRYPTO 2004, LNCS 3152, pp 443–459, 2004.
6. D. Boneh, X. Boyen, E. Goh, Hierarchical Identity Based Encryption with Constant Size Ciphertext, EUROCRYPT 2005, LNCS 3494, pp 440-456, 2005.
7. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. Journal Submission. Available from D. Boneh's website.
8. D. Boneh, M. Franklin. Identity Based Encryption from the Weil Pairing. CRYPTO 2001, LNCS 2139, pp. 213–229, 2001.

9. D. Boneh, M. Franklin. Identity Based Encryption from the Weil Pairing. SIAM J. of Computing, Vol. 32, No. 3, pp. 586–615, 2003.
10. D. Boneh and J. Katz. Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity Based Encryption. RSA-CT 2005, LNCS 3376, pp. 87-103, 2005.
11. R. Canetti, S. Halevi and J. Katz. A Forward-Secure Public-Key Encryption Scheme. EUROCRYPT 2003, LNCS 2656, pp 255-271. 2003.
12. R. Canetti, S. Halevi and J. Katz. Chosen-ciphertext Security from Identity Based Encryption. EUROCRYPT 2004. LNCS 3027, pp 207–222, 2004.
13. S. Galbraith, K. Harrison and D. Soldera. Implementing the Tate Pairing. ANTS V, LNCS 2369, pp. 324-337, 2002.
14. C. Gentry and A. Silverberg, Hierarchical ID-Based Cryptography, ASIACRYPT 2002, LNCS 2501, pp 548–566, 2002.
15. J. Horwitz and B. Lynn. Towards Hierarchical Identity-Based Encryption. EUROCRYPT 2002, LNCS 2332, pp 466–481, 2002.
16. A. Shamir. Identity-based Cryptosystems and Signature Schemes. CRYPTO 1984, LNCS 196, pp 47–53, 1985.
17. B. Waters. Efficient Identity-Based Encryption without Random Oracles. EUROCRYPT 2005, LNCS 3494, pp 114–127, 2005. Also available from Cryptology ePrint Archive, Report 2004/180, http://eprint.iacr.org/2004/180/.