Identity-Based Broadcasting

Yi Mu[1], Willy Susilo[1], and Yan-Xia Lin[2]

[1] School of IT and Computer Science
[2] School of Mathematics and Applied Statistics
University of Wollongong, Wollongong, NSW 2522, Australia
Email: {ymu,wsusilo,yanxia}@uow.edu.au

Abstract. In this paper, we introduce a new concept called "Identity-Based Broadcasting Encryption" (IBBE), which can be applied to dynamic key management in secure broadcasting. Based on this new concept, in the proposed system a broadcaster can *dynamically add* or *remove* a user to or from the receiver group without any involvement of users. We classify our systems into three different scenarios and give three provably secure and elegant constructions of IBBE system based on the pairing. Our system naturally suits *multi-group broadcasting*, where a message can be selectively broadcasted to certain groups of users.

Keywords: Broadcasting, Encryption, Pairing.

1 Introduction

Recently, a number of closely related models and constructions with the aim of securing electronic distribution of digital content have been proposed. An example of such services that rely on this kind of distribution is a pay TV system where a broadcaster needs assurance that only paid customers will receive the service and can only become viable if security of the distribution can be guaranteed.

The protection of a pay TV program is normally based on a symmetric-key cryptographic algorithm. That is, a broadcaster and all its users in a group share a secret key that is used by the broadcaster to encrypt a TV signal and is then used by users to decrypt the signal. The major disadvantage of such a scheme is that it is difficult for the broadcaster to stop an illegal user who has a forged secret key to receive pay TV programs. Changing a secret shared key requires updating all decoding boxes of users. This is infeasible and costly.

The key management problem above can be solved with a hybrid model. That is, the session key distribution relies on an efficient public-key algorithm that allows multiple decryption keys and the actual message is encrypted using the session key. With a dynamic key management, a broadcaster can *dynamically add* or *remove* a user to or from a receiver group without involvement of users. Conceptually, in such systems, an encryption key maps to multiple decryption keys (forming a set $\mathbb{K}$). A new decryption key can arbitrarily be added to $\mathbb{K}$ and an existing decryption key can arbitrarily be removed from $\mathbb{K}$ without involvement of users.

The concept of *secure broadcasting* was introduced by Fiat-Naor[5] for solving the problem of multi-message encryption, which is known as the *broadcast encryption.* Conceptually, a broadcast encryption is based on a single symmetric cipher equipped with a number of affine substitution boxes, where n messages can be converted into n ciphertexts that are broadcasted to the other end of a communication channel. The ciphertexts are then decrypted with the same key(s). We need to highlight that the broadcast encryption is completely different from our broadcasting concept that will be studied in this paper.

Another important related area is the work on *secure multicasting* [12]. In this concept, the multicast group must share a common key to enable the multicast communication. This problem is also known as the *re-keying* problem, which requires an algorithm to securely and efficiently update the group key whenever needed. Several constructions have been proposed (e.g. [12, 1, 10]) that consider the group's dynamic. However, we must point out that this system is not suitable for our purpose. In this system, each user needs to update his/her secret key whenever there is a dynamic in the system, for example due to the addition of a new user or removal of a user in the system. This solution is also not practical, since the user needs to update his/her secret key which might not be doable in some scenario (for example, consider a black box that is used to receive a pay TV broadcast channel).

In this paper, we introduce a new concept called "Identity-Based Broadcasting Encryption" (IBBE). Our schemes have the following distinct properties. (1) Users can be *dynamically* divided into groups with no involvement of users. (2) User groups can be dynamically *updated* by the broadcaster without any involvement of users. (3) It is *identity-based* so that the broadcaster can easily broadcast a message or messages to a group in term of the ID of the group. Furthermore, a group in our system can be divided into subgroups where each subgroup has a unique ID. The obvious application of sub-grouping is that a pay TV series is sold to a group while each subgroup could view a different program.

The primary reason to use the pairing in our system is that it allows most of computations to be done in elliptic curves and presents a promise for identity based encryption. The pairing such as the Weil pairing suggests that two points in an elliptic curve can be mapped to a point in a finite field. The Weil pairing was originally considered to be a bad thing, since it can be used for attacking elliptic curves[8]. Recently, it has been showed that the Weil pairing can be used to construct a protocol for three party one round Diffie-Hellman key exchange[7]. Boneh-Franklin have recently proposed a concrete identity based encryption protocol[3] and a short signature scheme based on the Weil pairing[4]. There have been a number of publications in the applications of the pairing. For example, Verheul has found the Weil pairing is useful for credential pseudonymous certificate systems[11]; Gentry and Silverberg introduced the concept of hierarchical ID-based cryptography using the Weil pairing in [6]; and Zhang and Kim proposed an ID-based signatures from pairing in [13].

We find that it is possible to use the Weil pairing to construct a mapping such that a public key can map into multiple private keys. These private keys can be dynamically split into multiple groups; each with a unique identity. Our schemes are instances of the elliptic curve discrete logarithm (ECDL) problems which are believed to be intractable.

It should be pointed out that a secure broadcasting scheme has been proposed to handle the dynamic user update issue [9]. In that system, a broadcaster can add/remove a user to/from any group dynamically. However, it is not identity-based and does not allow a group to have a subgroup in a broadcasting scenario. Moreover, the underlying assumption in that scheme is based on the intractability of the discrete logarithm problem and the overhead of the initial encryption key computation is proportional to the maximum number of users (although the computational overhead of encryption/decryption/update required is very low). Compared with [9], our new scheme shows better computational efficiency in the construction of initial encryption keys. This is due to the computation that is performed to "future" users that might join the system later on in [9]. In our schemes, the encryption key can be computed dynamically when a new user joins the system.

The rest of this paper is organized as follows. Section 2 gives the basic definitions and models of our systems. Section 3 provides some preliminaries that will be used in construction of our protocols. Section 4 describes the first IBBE scheme, where each group has a unique system setting and a group can be dynamically divided into subgroups. Section 5 presents the second scheme, where one set of cryptographic keys is required to broadcast a message to multiple groups, where each group has a group ID. Section 6 is devoted to a new protocol that is secure against exhaustive search attacks by insiders. Section 7 provides complete security proofs for all three protocols. Section 8 concludes the paper.

2 Definitions and models

In this section, we give the definitions and models of our systems.

Definition 1. *A designated group has an* ID *that is an arbitrary string* $\{0,1\}^*$.

Definition 2. *IBBE is a system that consists of a broadcaster* $\mathcal{T}$*, a set of* m *users* $\mathcal{U} = \{U_1, U_2, \cdots, U_m\}$*, and a set of* $\hat{k}$ *user groups* $\{\mathbb{U}^{\mathsf{ID}_1}, \mathbb{U}^{\mathsf{ID}_2}, \cdots, \mathbb{U}^{\mathsf{ID}_{\hat{k}}}\}$*. Each group* $\mathbb{U}^{\mathsf{ID}_i}$ *can contain* $\tilde{k}$ *subgroups* $\{\mathbb{U}_1^{\mathsf{ID}_i}, \mathbb{U}_2^{\mathsf{ID}_i}, \cdots, \mathbb{U}_{\tilde{k}}^{\mathsf{ID}_i}\}$ *or no subgroups. Each group* $\mathbb{U}^{\mathsf{ID}_i}$ *contains several users* $\subseteq \mathcal{U}$ *where each user* U_j *may belong to several group.*

The organization of groups is illustrated in Figure 1.

$\mathcal{T}$ has a private encryption key. Each user has a private decryption key. The management of keys varies in terms of group structures.

Definition 3. *Given a unique encryption key* $\mathcal{E}$ *(for* $\mathcal{T}$*) and users' decryption keys* $\mathcal{D}_i$*,* $\forall i \in \{1, 2, \cdots, m\}$*, the IBBE is referred to as the following scenarios:*

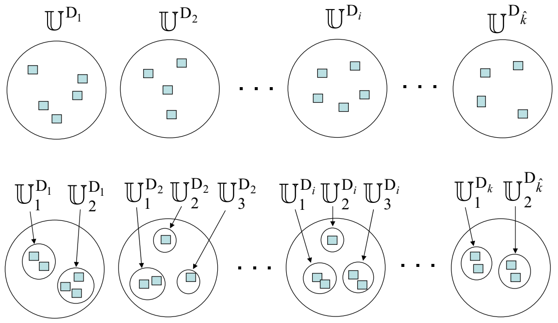
$\mathbb{U}^{\mathsf{D}_1}$
$\mathbb{U}^{\mathsf{D}_2}$
$\mathbb{U}^{\mathsf{D}_i}$
$\mathbb{U}^{\mathsf{D}_{\hat{k}}}$
. . .
. . .
$\mathbb{U}_1^{\mathsf{D}_1}$ $\mathbb{U}_2^{\mathsf{D}_1}$
$\mathbb{U}_1^{\mathsf{D}_2}$ $\mathbb{U}_2^{\mathsf{D}_2}$ $\mathbb{U}_3^{\mathsf{D}_2}$
$\mathbb{U}_1^{\mathsf{D}_i}$ $\mathbb{U}_2^{\mathsf{D}_i}$ $\mathbb{U}_3^{\mathsf{D}_i}$
$\mathbb{U}_1^{\mathsf{D}_k}$ $\mathbb{U}_2^{\mathsf{D}_{\hat{k}}}$

Fig. 1. The first row shows a set of user groups, where a user is represented by a square. Each group has a unique ID. In the second row, the user groups are divided into subgroups, where all subgroups have their parent's ID.

(1) *A group without subgroups. There exists a mapping Ψ_1 between a unique encryption key $\mathcal{E}_1$ and a group $\mathbb{U}^{\mathsf{ID}_i}$, $i \in \{1, ..., \hat{k}\}$. There exists a mapping Ψ_2 between the encryption key $\mathcal{E}_1$ and decryption keys $\mathcal{D}_i$, $i = 1, ..., m$, of users in $\mathbb{U}^{\mathsf{ID}_i}$, $i \in \{1, ..., \hat{k}\}$.*

(2) *A group with subgroups. There exists a mapping φ_1 between a unique encryption key $\mathcal{E}_2$ and a subgroup $\mathbb{U}_j^{\mathsf{ID}_i} \subset \mathbb{U}^{\mathsf{ID}_i}$, $j \in \{1, ..., \tilde{k}\}$. There exists a mapping φ_2 between the encryption key $\mathcal{E}_2$ and decryption keys $\mathcal{D}_i$, $i = 1, ..., k'$, of users in $\mathbb{U}_j^{\mathsf{ID}_i}$, where k' denotes the number of users in the subgroup.*

(3) *$\hat{k}$ groups without subgroups. There exists a mapping ω_1 between a unique encryption key $\mathcal{E}_3$ and groups $\mathbb{U}^{\mathsf{ID}_i}, \forall i \in \eta$, where η is a subset of $\{1, ..., \hat{k}\}$. There exists a mapping ω_2 between the encryption key $\mathcal{E}_3$ and decryption keys $\mathcal{D}_i$ of users in $\mathbb{U}^{\mathsf{ID}_i}, \forall i \in \eta$.*

In terms of the three scenarios, broadcasting can be classified as:

Scenario 1: A message encrypted with $\mathcal{E}_1$ can be decrypted by users in a single group $\mathbb{U}^{\mathsf{ID}_i}$, where ID_i is the ID of the corresponding group.
Scenario 2: A message encrypted with $\mathcal{E}_2$ can be decrypted by user in a subgroup $\mathbb{U}_j^{\mathsf{ID}_i}$ for the corresponding i and j.
Scenario 3: A message encrypted with $\mathcal{E}_3$ can be decrypted by users in groups $\mathbb{U}^{\mathsf{ID}_i}, \forall i \in \eta$.

Definition 4. *An* IBBE *is specified by five algorithms:* Setup, KeyGen, Encrypt, Decrypt, Update.

Setup : A randomized algorithm that takes as input a security parameter $\ell \in \mathbb{Z}$ and outputs system parameters (params1, params2). That is,

$$(\mathsf{params1}, \mathsf{params2}) \longleftarrow \mathsf{Setup}(\ell).$$

params1 is known only to the broadcaster $\mathcal{T}$. params2 is public.

KeyGen : A randomized algorithm that takes as input (ℓ, $\mathsf{params1}$, $\mathsf{params2}$), and outputs an encryption key tuple $\mathcal{E}$ and decryption keys $\mathcal{D}_i$ $(i = 1, ..., m)$. That is, $(\mathcal{E}, \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_m\}) \leftarrow \mathsf{KeyGen}(\ell, \mathsf{params1}, \mathsf{params2})$.
In Scenario 1, $\mathcal{E} = \mathcal{E}_1$, which is assigned to a user group $\mathbb{U}^{\mathsf{ID}_i}$.

$$\Psi_1 : \mathsf{ID}_i \mapsto \mathcal{E}_1, \quad \Psi_2 : \{\mathcal{D}_1, \cdots, \mathcal{D}_m\} \mapsto \mathcal{E}_1.$$

In Scenario 2, $\mathcal{E} = \{\mathcal{E}_{2,1}, \cdots, \mathcal{E}_{2,\tilde{k}}\}$, which are assigned to subgroups $\mathbb{U}_j^{\mathsf{ID}_i} \subset \mathbb{U}^{\mathsf{ID}_i}$, $j = 1, \cdots, \tilde{k}$.

$$\varphi_1 : \mathsf{ID}_i \mapsto \mathcal{E}, \quad \varphi_2 : \{\mathcal{D}_1, \cdots, \mathcal{D}_{k'}\} \mapsto \mathcal{E}_{2,j}, j \in \{1, \cdots, \tilde{k}\}.$$

In Scenario 3, $\mathcal{E} = \mathcal{E}_3$, which is assigned to groups $\mathbb{U}^{\mathsf{ID}_i}, i \in \eta$.

$$\omega_1 : \mathsf{ID}_i \mapsto \mathcal{E}_3, \forall i \in \eta, \quad \omega_2 : \mathcal{D}_i \mapsto \mathcal{E}_3, \forall i \in \eta.$$

$\mathsf{Encrypt}$: An algorithm that uses $\mathsf{params1}$, $\mathsf{params2}$, an encryption key $\mathcal{E}$, an ID, and a message M as its inputs and outputs a ciphertext tuple, c.

$$c \leftarrow \mathsf{Encrypt}(\mathsf{params1}, \mathsf{params2}, \mathcal{E}, \mathsf{ID}, M).$$

For clarity, in Scenario 1, the $\mathsf{Encrypt}$ algorithm is defined as

$$c \leftarrow \mathsf{Encrypt}(\mathsf{params1}, \mathsf{params2}, \mathcal{E}_1, \mathsf{ID}_i, M),$$

in Scenario 2, it is defined as

$$c \leftarrow \mathsf{Encrypt}(\mathsf{params1}, \mathsf{params2}, \mathcal{E}_{2,j}, \mathsf{ID}_i, M), \quad \forall j \in \{1, \cdots, \tilde{k}\}$$

and in Scenario 3, it is defined as

$$c \leftarrow \mathsf{Encrypt}(\mathsf{params1}, \mathsf{params2}, \mathcal{E}_3, \mathsf{ID}_{\forall i \in \eta}, M).$$

$\mathsf{Decrypt}$: An algorithm that takes as input: $\mathsf{params2}$, one of decryption keys $\mathcal{D}_i$, $i = 1, 2, \cdots, m$, and a ciphertext c, and outputs the corresponding plaintext M, if $\mathcal{D}_i$ is valid. It outputs $\perp$ otherwise.

$$\mathsf{Decrypt}(\mathsf{params2}, \mathcal{D}_i, c) = \begin{cases} M & \text{if } \mathcal{D}_i \text{ is valid} \\ \perp & \text{otherwise} \end{cases}$$

Update : An algorithm that takes as input the encryption key tuple, and outputs a new encryption key tuple. That is,

$$\hat{\mathcal{E}} \leftarrow \mathsf{Update}(\mathcal{E})$$

The basic setup of our schemes are based on the pairing. The basic system parameters for our systems are described as follows. Let E denote an elliptic curve over a field K with characteristic > 0, and $E[n]$ be its group of n-torsion points.

Definition 5. *Let $n \in \mathbb{Z}_{\geq 2}$ denote an integer, coprime to the characteristic of K with characteristic > 0. The Weil pairing is a mapping*

$$\hat{e} : E[n] \times E[n] \to \mu_n$$

where μ_n is the group of nth roots of unity in $\bar{K}$.

Under the definition of the Weil pairing, if $\hat{e}(P, Q)$ is not the unit in μ_n, then $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for $P, Q \in E[n]$ and all $a, b \in \mathbb{Z}$. Please refer to Page 43 of [2] for details of the Weil pairing.

Group $E[n]$ is a cyclic additive group, now denoted $\mathbb{G}_1$, which maps to a cyclic multiplicative group $\mathbb{G}_2$ by the Weil pairing. If n is prime, then both G_1 and G_2 have a prime order. From Definition 5, n is not necessary to be prime. Thus, the order of $\mathbb{G}_1$ and $\mathbb{G}_2$ is not necessarily prime. In this paper, we consider the case where the order q of $\mathbb{G}_1$ and $\mathbb{G}_2$ is a product of some primes. q is kept secret by the broadcaster, since users do not need it in decryption. For simplicity, we will omit modulus n in the presentation.

For convenience of the presentation, hereafter we will denote the pairing $\hat{e}$ by $\langle ., . \rangle$.

3 Preliminaries

Before describing our schemes, in this section we give some basic results to support the validity and practicability of our schemes.

Definition 6. *An integer u_i is called an Identity Element associated with u'_i and q, defined by $I(u'_i, q)$, if the following property is held: $u_i u'_i = u'_i \bmod q$.*

Lemma 1. *Assume $q = p_1^{k_1} p_2^{k_2} \cdots p_T^{k_T}$, where p_i are primes and k_i are integers ($p_i \neq p_{i'}$ for $i \neq i'$). Set $u'_i \leftarrow p_i^{k_i}$, $i = 1, ..., t$ and $t < T$. Set $u_i \leftarrow \prod_{j \neq i} p_j^{k_j} + 1$. Then, u_i is an $I(u'_i, q)$. Also, there exists no u'_{j_0} for $j_0 \neq i$, such that u_i is an $I(u'_{j_0}, q)$.*

Proof. Proving that u_i is an $I(u'_i, q)$ is equivalent to proving $(u_i - 1)u'_i = kq$ for certain k. By noting that $(u_i - 1)u'_i = (\prod_{i \neq j} p_j^{k_j}) p_i^{k_i} = q$, it is obvious that $u_i u'_i = u'_i \bmod q$ is held.

We prove the second statement by contradiction. If there is u'_{j_0} such that $u_i u'_{j_0} = u'_{j_0} \bmod q$, $i \neq j_0$, then there exists an integer $k_0 \neq 0$ such that $(u_i - 1)u'_{j_0} = k_0 q$. This implies $\prod_{j \neq i} p_j^{k_j} p_{j_0}^{k_{j_0}} = k_0 \prod_{j=1}^{T} p_j^{k_j}$. That is, $p_{j_0}^{k_{j_0}} = k_0 p_i^{k_i}$. It is contradictory to which p_{j_0} is prime, as $p_{j_0} \neq p_i$ and $k_0 \neq 0$. □

Remark 1: Given u'_i as above, there may exist more than one $I(u'_i, q)$. For example, $\prod_{j \neq i} p_j^{k_j} p_i + 1$ is also an $I(u'_i, q)$. However, using u_i defined in the lemma above, we ensure that u_i is unique to $I(u'_i, q)$.

Definition 7. *The doublet (u_i, u'_i) given in Lemma 1 is defined as a "qualified pair."*

Definition 8. *Let $v \in Z_q$ be prime and* $\gcd(v,q) = 1$*. An integer $v_i \in Z_q$ is called the image of u_i associated with v if $vv_i = u_i \ mod q$.*

Lemma 2. *The mapping from u_i to its image v_i is unique.*

Proof. Since $\gcd(q,v) = 1$, v has an inverse, v^{-1}. v_i can then be computed from

$$v_i = v^{-1} u_i \bmod q, \qquad i = 1, 2, \cdots, m.$$

Thus,

$$vv_i \bmod q = (vv^{-1}u_i) \bmod q = u_i.$$

It is trivial that $v_i \neq v_j$ if $u_i \neq u_j$. □

4 Identity Based Broadcasting

This scheme fits into Scenario 1 given in Section 2. We assume that users are assigned to multiple groups. Each group $\mathbb{U}^{\mathsf{ID}_i}$, $i \in \{1, ..., \hat{k}\}$, has a unique identity (ID). In terms of security, we require the broadcaster $\mathcal{T}$ to have a separate ID-based encryption key for each group. Each user in a group is assigned a unique decryption key. An encrypted message broadcasted to a targeted group can be decrypted by any of users in the group. Group members can be dynamically added to or removed from a group by $\mathcal{T}$ without involvement of any existing users. We here use the words "encryption key" to replace "public key", because the encryption key (tuple) is only known to $\mathcal{T}$.

4.1 The protocol

Setup : $\mathcal{T}$ inputs $\ell \in \mathbb{Z}$ as a security parameter to generate private params1 $\leftarrow (P \in \mathbb{G}_1, q)$ and public params2$\leftarrow (\mathbb{G}_1, \mathbb{G}_2, \hat{e})$ as output. He then constructs two strong hash functions, $H_1 : \{0,1\}^l \rightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^l$. H_1 is only known to $\mathcal{T}$ and H_2 is publicly available.

KeyGen : $\mathcal{T}$ inputs (ℓ, params1, params2) to KeyGen and obtains

- $u \leftarrow \prod_{i=1}^{m} u'_i \bmod q$, where $u_i u'_i = u'_i \bmod q$. Namely, (u_i, u'_i) is a qualified pair defined earlier.
- a prime v such that $\gcd(v,q) = 1$,
- $\{v_i\}$ as images of $\{u_i\}$ such that $vv_i \bmod q = u_i$, and
- a set of keys described as follows:
  - Set $d_i \leftarrow (xu_i + 1)v_i \bmod q$, where x, the master key, is a prime selected from $\mathbb{Z}_q$.
  - Set $E_1 \leftarrow uP$ and $E_2 \leftarrow uvP$.
  - Extract the group ID from the group identifier $\mathsf{ID} \in \{0,1\}^l$: $Q_{\mathsf{ID}} \leftarrow H_1(\mathsf{ID})$. The decryption key for a user is $D_i \leftarrow d_i Q_{\mathsf{ID}}$.

The outputs are the encryption key triplet (E_1, E_2, x) and decryption keys D_i, $i = 1, \cdots, m$. We note that the only information for U_i will be his decryption key D_i, and the public modulus n along with $\hat{e}, \mathbb{G}_1, \mathbb{G}_2$. All other data including P, Q_{ID} are known to $\mathcal{T}$ only.

Encrypt : $\mathcal{T}$ inputs a message $M \in \{0,1\}^l$ and the encryption key tuple (E_1, E_2, x), selects a random $r \in \mathbb{Z}_q$ and then computes $R \leftarrow rE_2$, and $b \leftarrow \langle E_1, (x+1)Q_{\mathsf{ID}} \rangle$. The ciphertext is obtained from a bitwise XOR operation $c \leftarrow M \oplus H_2(b^r)$. The output from Encrypt is the tuple: (c, R), which is then broadcasted to users in the group with the group ID: ID_i. Note that M could be a session key and the real message is encrypted with this session key.

Decrypt : $U_i \in \mathbb{U}^{\mathsf{ID}_i}$ inputs (c, R) and his decryption key D_i and computes

$$\begin{aligned} \langle R, D_i \rangle &= \langle rvuP, (xu_i+1)v_i Q_{\mathsf{ID}} \rangle \\ &= \langle ruP, (xu_i+1)u_i Q_{\mathsf{ID}} \rangle \\ &= \langle rE_1, (x+1)Q_{\mathsf{ID}} \rangle \\ &= b^r. \end{aligned}$$

Upon obtaining this value, he can decrypt the message $M \leftarrow c \oplus H_2(b^r)$.

Remark 2. The broadcaster $\mathcal{T}$ can arbitrarily construct user groups at a runtime so that each group can receive a single pay TV program they entitle to watch (Scenario 2). It is actually trivial to achieve it in our protocol. What $\mathcal{T}$ has to do is to construct a new u that is the product of u_j for $j \in \{\text{selected users}\}$. That is, $u \leftarrow \prod_j u'_j \bmod q$.

Lemma 3. *The collusion of t users in the system, $t \leq m$, cannot produce a valid decryption key.*

Proof. We are interested to see what can be gained by a collusion of t users in the system. It is clear to see that t users cannot gain a new decryption key. Without loss of generality, we assume that there are two malicious users who hold two legitimate decryption keys D_A and D_B respectively. By addition, they can obtain $D' \leftarrow D_A + D_B$, hoping that D' is a new decryption key. However, a "decryption" with D' will produce $(b^r)^2$. Since q is unknown to users, it is infeasible for users to compute the inverse of 2 in order to remove 2. It is easy to check other cases; therefore, we here omit the details. □

The complete security proof of this scheme is given in Section 7.

4.2 Dynamic Update

The idea is to allow $\mathcal{T}$ to add a new user U_z to or remove an existing user $U_{z'}$ from the system without current users' involvement. Formally, we allow

$$\hat{\mathbb{U}}^{\mathsf{ID}_i} \leftarrow \mathbb{U}^{\mathsf{ID}_i} \cup \{U_z\}$$

and

$$\hat{\mathbb{U}}^{\mathsf{ID}_i} \leftarrow \mathbb{U}^{\mathsf{ID}_i} \setminus \{U_{z'}\}$$

without any involvement of $\forall U_j \in \mathbb{U}^{\mathsf{ID}_i}$.

When a user $U_{z'} \in \mathbb{U}^{\mathsf{ID}_i}$ is to be removed from the current system, $\mathcal{T}$ simply updates the encryption key by recomputing his encryption key as $\hat{E}_1 \leftarrow u_{z'}^{-1}E_1$, $\hat{E}_2 \leftarrow u_{z'}^{-1}E_2$ ($\hat{E}_1, \hat{E}_2$ are now the new encryption key tuple).

Adding a new user $U_z \notin \mathbb{U}^{\mathsf{ID}_i}$ into a group can be done with a similar fashion: $\hat{E}_1 \leftarrow u_z E_1$, $\hat{E}_2 \leftarrow u_z E_2$. The update scheme mentioned in this section is also applicable to the next two schemes that will be discussed in the next sections.

5 Broadcast to multiple groups

In the preceding scheme, a message can be broadcasted to multiple groups, but it requires a separate encryption key for each group and a message has to be encrypted several times. In this section, we will describe a new approach (Scenario 3) that has the following important features.

- A message can be broadcasted to multiple groups without multiple encryptions.
- $\mathcal{T}$ can use a specified ID for each group in the broadcast message.
- $\mathcal{T}$ can still use the original encryption key defined in the previous section.
- The protocol naturally has chosen-ciphtertext security.

We assume that the total number of users is m. They are divided into $\hat{k}$ groups, namely $\mathbb{U}^{\mathsf{ID}_1}, \mathbb{U}^{\mathsf{ID}_2}, \cdots, \mathbb{U}^{\mathsf{ID}_{\hat{k}}}$. As in the preceding protocol, the secret decryption key for a user $U_i \in \mathbb{U}^{\mathsf{ID}_j}$ is denoted by $D_i^{\mathsf{ID}_j}$.

The encryption scheme is similar to that in the preceding protocol. The only required change is to construct the encryption key with respect to all group IDs. For clarity, for each group in $\{\mathbb{U}^{\mathsf{ID}_1}, \mathbb{U}^{\mathsf{ID}_2}, \cdots, \mathbb{U}^{\mathsf{ID}_{\hat{k}}}\}$, we rewrite b (as defined in the preceding scheme) as $b_1, b_2, \cdots b_{\hat{k}}$, respectively. They are still computed from the the formula: $b_X \leftarrow \langle E_1, (x+1)Q_X \rangle$ for $\mathbb{U}^X$.

Without loss of generality, suppose that a pay TV program is intended to broadcast to groups $\mathbb{U}^{\mathsf{ID}_1}$ and $\mathbb{U}^{\mathsf{ID}_2}$, then $X \in \{\mathsf{ID}_1, \mathsf{ID}_2\}$. $\mathcal{T}$ needs to compute a new parameter as part of key distribution for each group as follows. Select $b'_{\mathsf{ID}_1}, b'_{\mathsf{ID}_2}$ such that they satisfy $b_X b'_X = \hat{b}_X$ for $\mathbb{U}^X$.

$\mathsf{Encrypt}$: $\mathcal{T}$ carries out the following procedures:

- inputs a message $M \in \{0,1\}^l$, the encryption key E_1, E_2, x, and the session key $\hat{b}_X$ for $X \in \{\mathsf{ID}_1, \mathsf{ID}_2\}$,
- selects three additional cryptographic hash functions: $H_3 : \{0,1\}^l \times \{0,1\}^l \rightarrow \mathbb{Z}_q$, $H_4 : G_2 \rightarrow G_2$, and $H_5 : G_2 \rightarrow \{0,1\}^l$, where H_4 is publicly available and H_3, H_5 are known to $\mathcal{T}$ only.
- sets $r \leftarrow H_3(\sigma, M)$, where σ is selected at random,
- computes $R \leftarrow rE_2$, and $b_X \leftarrow \langle E_1, (x+1)Q_X \rangle$, where $X \in \{\mathsf{ID}_1, \mathsf{ID}_2\}$,

- computes $c_X \leftarrow {b'}^r_X H_4(b^r_X)$, and
- computes the ciphertext from a bitwise XOR operation $c \leftarrow M \oplus H_2(\hat{b}^r_X)$.
  Output: $(c_X,\ c,\ R)$, which is then broadcasted.

$\mathsf{Decrypt}$: A user $U_i \in \mathbb{U}^{\mathsf{ID}_1}$ or $\mathbb{U}^{\mathsf{ID}_2}$ inputs $(c,\ c_X,\ R,\ D_i^{(X)})$ for $X \in \{\mathsf{ID}_1, \mathsf{ID}_2\}$ to the decryption algorithm and computes

$$\begin{aligned}\langle R, D_i^{(X)} \rangle &= \langle rvuP, (xu_i + 1)v_iQ_X \rangle \\ &= \langle ruP, (x+1)Q_X \rangle \\ &= b^r_X,\end{aligned}$$

and obtains $c_X(H_4(b^r_X))^{-1} = {b'}^r_X$ and ${b'}^r_X b^r_X = \hat{b}^r_X$. Revealing this value, he can obtain the message $M \leftarrow c \oplus H_2(\hat{b}^r)$.

This protocol is chosen-ciphertext secure, since we have used the technique due to Boneh-Franklin [3]. The security proof is omitted. The reader is referred to [3] for details. The security against the chosen plaintext attacks is given in Section 7.

6 A Protocol against Exhaustive Search

We now describe an IBBE that is secure against exhaustive search by a legitimate user who wishes to find another decryption key pair, which is different from his own. This scheme is suitable for both cases described in Sections 4 and 5. We take the first one in the following description.

KeyGen: The basic setup is the same as that of the first protocol. $\mathcal{T}$ needs to reconstruct decryption key pairs. For a user $U_i \in \mathbb{U}^{\mathsf{ID}}$, a private key pair is constructed as $D_i \leftarrow x(u_i + y_i)v_iQ_{\mathsf{ID}}$, $d_i \leftarrow (1 + y_i)^{-1} \bmod q$, where $x \in \mathbb{Z}_q$ is the master key and y_i is an appropriate integer in $\mathbb{Z}_q$ such that $(1 + y_i)$ has an inverse in $\mathbb{Z}_q^*$. y_i, u_i, v_i are unique to U_i and x is unique to the system.

$\mathsf{Encrypt}$: To encrypt a message $M \in \{0,1\}^l$, $\mathcal{T}$ chooses a number $r \in \mathbb{Z}_q$ and sets $R \leftarrow rE_2$. The ciphertext c is constructed from $M \oplus H_2(b^r)$, where $b = \langle E_1, xQ_{\mathsf{ID}} \rangle$.

$\mathsf{Decrypt}$: $U_i \in \mathbb{U}^{\mathsf{ID}}$ computes

$$\langle R, D_i \rangle = \langle ruvP, x(u_i + y_i)v_iQ_{\mathsf{ID}} \rangle = \langle uP, xQ_{\mathsf{ID}} \rangle^{r(1+y_i)} \equiv \hat{b}_i,$$

$$\hat{b}_i^{d_i} = b^r.$$

Hence, he can reveal M by computing $M \leftarrow c \oplus H_2(b^r)$.

Lemma 4. *It is infeasible for $U_i \in \mathbb{U}^{\mathsf{ID}}$ to find a decryption key tuple that is different from his own decryption key, in terms of a ciphertext and his own decryption key.*

Proof. We prove it by contradiction. Assume that U_i is able to create a different decryption key tuple. Assume U_i has found a tuple of decryption key, $D \in \mathbb{G}_1$ and $d \in Z_q$, where D, d are independent of U_i's personal decryption key (D_i, d_i). Since $c = M \oplus H_2(b^r)$ and R are public, D_i and d_i must satisfy the following equations: $\langle R, D_i \rangle = \hat{b}_i$ and $\hat{b}_i^{d_i \bmod q} = b^r$, where $\hat{b}_i$ and b^r are known since U_i can use his own key pair to obtain the information of $\hat{b}_i$ and b^r.

Observing $\langle uP, xQ_{\mathsf{ID}} \rangle^{r(1+y_i)} = \hat{b}_i$, in order to obtain

$$\hat{b}_i^{d \bmod q} = \langle uP, xQ_{\mathsf{ID}} \rangle^{r(1+y_i)d} = b^r = \langle uP, xQ_{\mathsf{ID}} \rangle^r$$

d must satisfy $(1 + y_i)d = 1 \bmod q$. This suggests that $d = (1 + y_i)^{-1} \bmod q$ and U_i's key tuple is found. We obtain the contradiction. □

Lemma 5. *The collusion of t users in the system, $t \leq m$, cannot produce a valid decryption key that can be given to a malicious user.*

Proof. Similar to the proof of Lemma 3.

7 Security Consideration

Consider an IBBE with encryption algorithm $\mathcal{S}$ wrt $M \oplus H_2(b^r)$, where M is a true message. We denote a ciphertext by c_S. We will omit the subscript S when it is clear from the context. Define a system $(K_{pub}, K_{pri}, \mathcal{L}, H_2)$, where K_{pub} contains all public information; K_{pri} consists of all private keys (encryption and decryption keys) and private hash functions; $\mathcal{L}$ is an operator mapping from (K_{pub}, K_{pri}) into $\mathbb{G}_2$; H_2 is a random oracle from $\mathbb{G}_2$ to $\{0,1\}^l$.

Definition 9. *If the output $O \leftarrow \mathcal{L}(K_{pub}, K_{pri})$ is independent of K_{pri} and, for any c_S, $c_S \oplus H_2(O)$ outputs the true message M, we will call the system wrt $(K_{pub}, K_{pri}, \mathcal{L}, H_2)$ a* Valid *IBBE system associated to $\mathcal{S}$ through the decryption procedure $c_S \oplus H_2(O)$.*

(1) Consider the first scheme in Section 4. Let K_{pub} be a set consisting of $ruvP$ and $ruy'P$; K_{pri} consist of D and D' and operation $\mathcal{L}$ be defined as

$$\mathcal{L}(K_{pub}, K_{pri}) = \langle ruvP, D \rangle.$$

Then $(K_{pub}, K_{pri}, \mathcal{L}, H_2)$ forms a Valid system, because for any valid private key D, we have $\mathcal{L}(K_{pub}, K_{pri}) = b^r$ which is independent of any valid private key D and, for any encrypted message C given by the scheme, we have $c \oplus H_2(b^r) = M$, which is the true message.

(2) Consider the second scheme in Section 5. Let K_{pub} be a set consisting of (R, H_4, c_X); K_{pri} be $D^{(X)}$; and operation $\mathcal{L}$ be defined as

$$\mathcal{L}(K_{pub}, K_{pri}) = c_X(H_4(\langle R, D^{(X)} \rangle))^{-1} \langle R, D^{(X)} \rangle$$

Then $(K_{pub}, K_{pri}, \mathcal{L}, H_2)$ forms a Valid system, because for any valid private key $D^{(X)}$, $\mathcal{L}(K_{pub}, K_{pri}) = b^r$ which is independent of any valid private key $D^{(X)}$ and, for any ciphertext C of the scheme, we have $C \oplus H_2(b^r) = M$, which is the true message.

(3) Consider the third scheme in Section 6. Let K_{pub} be a set of (R, H_2, c); K_{pri} be a set of D and d, and $\mathcal{L}$ be defined as:

$$\mathcal{L}(K_{pub}, K_{pri}) = \langle R, D \rangle^d,$$

then, $(K_{pub}, K_{pri}, \mathcal{L}, H_2)$ forms a Valid system, because for any valid private key pair (D,d), $\mathcal{L}(K_{pub}, K_{pri}) = b^r$, and for any ciphertext c of the scheme, $c \oplus H_2(b^r) = M$ gives the true message.

Lemma 6. *Let $\mathcal{S}$ be an encryption algorithm wrt $M \oplus H_2(b^r)$ that maps a ciphertext string into $\{0,1\}^l$. Assume that $(K_{pub}, K_{pri}, \mathcal{L}, H_2)$ is a* Valid *IBBE system associated with $\mathcal{S}$. If there is an adversary $\mathcal{A}$ with advantage ε_{N_h} against $\mathcal{S}$ after making a total of $N_h > 0$ queries to H_2, then there is an algorithm $\mathcal{B}$ with advantage at least $\frac{2^l \epsilon_{N_h} - 1}{2^l - 1}$ for identifying a valid decryption key with the running time is $\mathcal{O}(time\ (\mathcal{A}))$.*

Proof. For a given Valid IBBE system $(K_{pub}, K_{pri}, \mathcal{L}, H_2)$ with the encryption algorithm $\mathcal{S}$, we first define the algorithm $\mathcal{B}$ and, then, prove that the advantage of $\mathcal{B}$ taking into account the advantage of $\mathcal{A}$.

The input to $\mathcal{B}$ is K_{pub}. $\mathcal{B}$ picks a random string $\tilde{c}$ from $\{0,1\}^l$ and assumes that $\tilde{c}$ is an encrypted message. That is, there is a true message M such that $\tilde{c} = M \oplus H_2(\mathcal{L}(K_{pub}, \tilde{K}_{pri}))$. Let $\mathbb{K}_{pri}$ be a set contains all potential $\tilde{K}_{pri}$ for the underlying IBBE system. Obviously, the size of this set is very large and the likelihood that randomly picking up an element from $\mathbb{K}_{pri}$ that is a valid K_{pri} for the system is negligible. Otherwise, the following study is meaningless.

Challenge: $\mathcal{B}$ randomly chooses an element from $\mathbb{K}_{pri}$ to form a $\tilde{K}_{pri}$, and then, sends K_{pub}, $\tilde{K}_{pri}$, and $\tilde{c}$ to $\mathcal{A}$. $\mathcal{B}$ wants to utilize $\mathcal{A}$'s knowledge to make a decision if this $\tilde{K}_{pri}$ can be accepted as a valid key. For convenience, we call $\tilde{K}_{pri}$ as the candidate of a valid decryption key.

H_2**-queries:** B will independently repeat the above challenge N_h times and obtain a list of candidate keys, say $\{\tilde{K}_{pri,i}\}$. At the same time, $\mathcal{A}$ independently quires H_2 for N_h times based on $\mathcal{B}$'s requirement and observes the outputs

$$\tilde{c} \oplus H_2(\mathcal{L}(K_{pub}, \tilde{K}_{pri,i}) = \hat{M}_i, \qquad i = 1, 2, \cdots, N_h.$$

$\mathcal{B}$ will establish a list with all these outputs. The list is denoted by H_{list} having elements $\{(\tilde{K}_{pri,i}, \hat{M}_i)\}$.

Guess: After the N_h queries, $\mathcal{A}$ makes a guess on the true message M, say $\hat{M}$. If $\hat{M}$ coincides with some $\hat{M}_i$, say a $\hat{M}_{i_0}$, in the list H_{list}, $\mathcal{B}$ will consider $(\tilde{K}_{pri,i_0})$ as a valid key; if $\hat{M}$ does not appear in H_{list}, $\mathcal{B}$ then randomly picks a key from $\mathbb{K}_{pri}$ and assigns it as K_{pri}.

We now show that, if $\mathcal{B}$ uses the above procedure to guess a valid K_{pri}, the probability of obtaining a really valid K_{pri} is at least $\frac{2^l \epsilon_{N_h} - 1}{2^l - 1}$. For convenience, we also denote by H_{list} the event that at least one valid $\tilde{K}_{pri}$ appears in the list H_{list}. Denote $\mathcal{B}_M$ the event that, after N_h quires, a valid $\tilde{K}_{pri}$ is identified through the above procedure. Then, we have

$$\begin{aligned} P(\mathcal{B}_M) &= P(H_{list})P(\mathcal{B}_M|H_{list}) + (1 - P(H_{list}))P(\mathcal{B}_M|H^c_{list}) \\ &\geq P(H_{list})P(\mathcal{B}_M|H_{list}) = P(H_{list}), \end{aligned}$$

where H^c_{list} denotes the complement of H_{list}.

Since after N_h quires, the probability of $\mathcal{A}$ obtaining the true message is at least ϵ_{N_h},

$$P(\hat{M} = M) > \epsilon_{N_h}.$$

Thus

$$\begin{aligned} \varepsilon_{N_h} < P(\hat{M} = M) &= P(H_{list})P(\hat{M} = M|H_{list}) + (1 - P(H_{list}))P(\hat{M} = M|H^c_{list}) \\ &= P(H_{list}) + \frac{1}{2^l}(1 - P(H_{list})), \end{aligned}$$

and

$$P(H_{list}) > \frac{2^l\epsilon_{N_h} - 1}{2^l - 1}.$$

This gives $P(B_M) > \frac{2^l\epsilon_{N_h}-1}{2^l-1}$. □

Corollary 1 *If there is an adversary $\mathcal{A}$ with advantage ϵ_{N_h} against $\mathcal{S}$ after making a total of $N_h > 0$ queries to H_2, then for Scheme 1, there is an algorithm $\mathcal{B}$ such that finding valid pair keys D and D' with advantage at least $\frac{2^l\epsilon_{N_h}-1}{2^l-1}$ and the running time is $\mathcal{O}(time\ (\mathcal{A}))$.*

Corollary 2 *If there is an adversary $\mathcal{A}$ with advantage ϵ_{N_h} against $\mathcal{S}$ after making a total of $N_h > 0$ queries to H_2, then for Scheme 2, there is an algorithm $\mathcal{B}$ such that finding valid pair keys $D^{(X)}$ and $D'^{(X)}$ with advantage at least $\frac{2^l\epsilon_{N_h}-1}{2^l-1}$ and a running time is $\mathcal{O}(time\ (\mathcal{A}))$.*

Corollary 3 *If there is an adversary $\mathcal{A}$ with advantage ϵ_{N_h} against $\mathcal{S}$ after making a total of $N_h > 0$ queries to H_2, then for Scheme 3, there is an algorithm $\mathcal{B}$ such that finding valid pair keys D and d with advantage at least $\frac{2^l\epsilon_{N_h}-1}{2^l-1}$ and the running time is $\mathcal{O}(time\ (\mathcal{A}))$.*

8 Conclusion

We formalized the model of IBBE scheme. We proposed three identity based broadcast schemes that meet the requirement of IBBE. In these systems, a user group can be dynamically updated by the broadcaster without any involvement of any other users. The algorithm for updating the group is simple and efficient, since the broadcaster is not required to recompute the entire encryption key. These schemes are proven to be secure; especially the third protocol that is secure against exhaustive research attacks. We provided a complete security proof for our schemes.

References

1. J. Anzai, N. Matsuzaki, and T. Matsumoto. A Quick Group Key Distribution Scheme with "Entity Revocation". *Advances in Cryptology - Asiacrypt '99, Lecture Notes in Computer Science 1716*, pages 333 – 347, 1999.
2. I. Blake, G. Seroussi, and N. Smart. *Elliptic Curves in Cryptography.* Cambridge Unversity Press, 2001.
3. D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology, Crypto 2001,* Lecture Notes in Computer Science 2139, pages 213–229. Springer Verlag, 2001.
4. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In *Advances in Cryptology, Asiacrypt 2001, Lecture Notes in Computer Science 2248*, pages 514–532. Springer Verlag, 2001.
5. A. Fiat and M. Naor. Broadcast encryption. In *Advances in Cryptology, Crypto '93,* Lecture Notes in Computer Science 773, pages 480–491. Springer Verlag, 1994.
6. C. Gentry and A. Silverberg. Hierarchical ID-based Cryptography. *Advances in Cryptology, Asiacrypt 2002, Lecture Notes in Computer Science 2501*, pages 548 – 566, 2002.
7. A. Joux. A One Round Protocol for Tripartite Diffie-Hellman. In W. Bosma, editor, *ANTS-IV, Lecture Notes in Computer Science*, pages 385–394. Springer Verlag, 2000.
8. A. Menezes, T. Okamoto, and S. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transaction on Information Theory*, 39:1639–1646, 1993.
9. Y. Mu and V. Varadharajan. Robust and secure broadcasting. In *Indocrypt 2001, Lecture Notes in Computer Science*, pages 223 – 231. Springer, 2001. (the revised version: www.uow.edu.au/˜ymu).
10. M. Naor and B. Pinkas. Efficient Trace and Revoke Schemes. *Financial Cryptography 2000, Lecture Notes in Computer Science 1962*, pages 1– 20, 2001.
11. E. R. Verheul. Self-blindable credential certificates from the weil pairing. In *Advances in Cryptology–Asiacrypt 2001,* Lecture Notes in Computer Science 2248, pages 533–551. Springer Verlag, 2001.
12. D. M. Wallner, E. J. Harder, and R. C. Agee. Key management for multicast: Issues and architectures. Internet Draft (draft-wallner-key-arch-01.txt), ftp://ftp.ietf.org/ internet-drafts/ draft-wallner-key-arch-01.txt.
13. F. Zhang and K. Kim. ID-based Blind Signature and Ring Signature from Pairings. *Advances in Cryptology, Asiacrypt 2002, Lecture Notes in Computer Science 2501*, pages 33– 547, 2002.